

MENU

SEARCH

INDEX

DETAIL

JAPANESE

BACK

NEXT

7 / 18

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-194655
 (43)Date of publication of application : 14.07.2000

(51)Int.Cl. G06F 13/38
 G06F 13/00
 H04L 12/56
 H04L 13/08

(21)Application number : 11-277881 (71)Applicant : STMICROELECTRONICS INC
 (22)Date of filing : 30.09.1999 (72)Inventor : CHRISTIAN D CASPER

(30)Priority

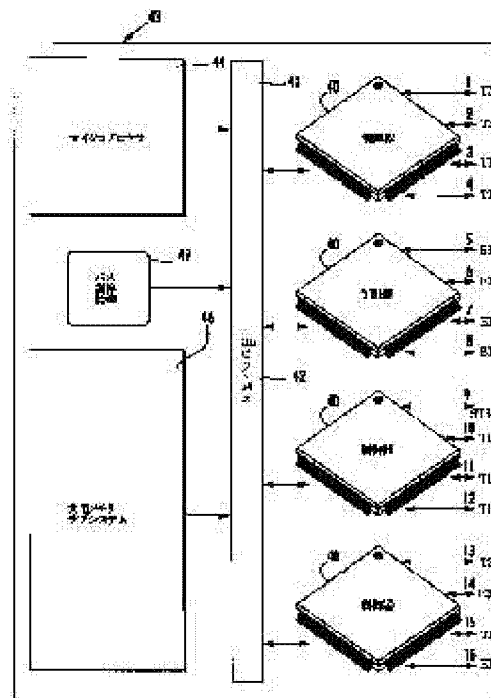
Priority number : 98 163952 Priority date : 30.09.1998 Priority country : US

(54) METHOD AND SYSTEM FOR TRANSFER CONTROL OF DATA BY UPDATING DESCRIPTOR IN DESCRIPTOR RING

(57)Abstract:

PROBLEM TO BE SOLVED: To reduce the overhead regarding the use of descriptors and descriptor rings relating to respective frame data buffers by updating only the starting and ending descriptors in a descriptor chain for a desired host or controller.

SOLUTION: A network controller 40 updates a starting and an ending descriptor entry which are concerned to optimize the use of a bus when ≥ 3 frame data buffers are chained together. When the network controller 40 completes the process as to a buffer relating to chained frames, the possession of the ending descriptor is returned first and then the possession of the starting descriptor is returned. The flags and fields of the starting and ending descriptors in a column type chain of a fence post, i.e., a fence are all updated by the controller 40 and when they are completely sent or received, accurate information regarding the frames is provided.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号
特開2000-194655
(P2000-194655A)

(43)公開日 平成12年7月14日(2000.7.14)

(51)Int.Cl. ⁷	識別記号	F I	テームコード* (参考)
G 0 6 F 13/38	3 1 0	G 0 6 F 13/38	3 1 0 B
13/00	3 5 3	13/00	3 5 3 A
H 0 4 L 12/56		H 0 4 L 13/08	
13/08		11/20	1 0 2 Z

審査請求 未請求 請求項の数42 O L (全 48 頁)

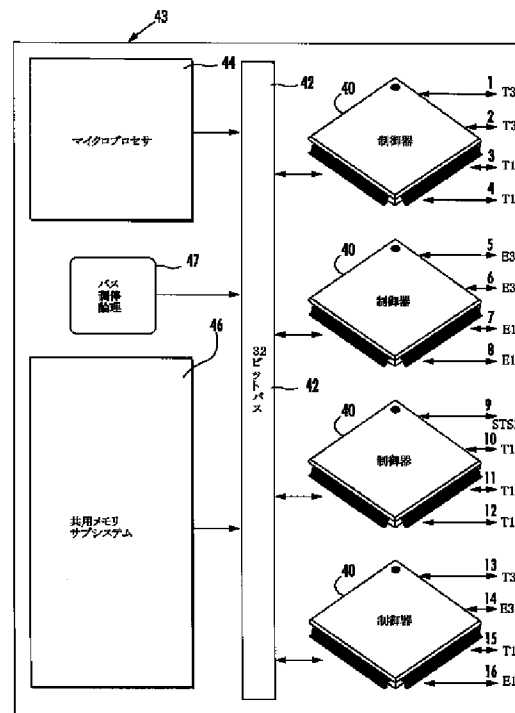
(21)出願番号	特願平11-277881	(71)出願人	591236448 エスディーマイクロエレクトロニクス、インコーポレイテッド STMicroelectronics, Inc アメリカ合衆国、 テキサス 75006, カーロルトン、 エレクトロニクス ドライ ブ 1310
(22)出願日	平成11年9月30日(1999.9.30)	(72)発明者	クリスチャン デイ、 キャスパ アメリカ合衆国、 テキサス 75007, カーロルトン、 オールド デントン 3640
(31)優先権主張番号	0 9 / 1 6 3 9 5 2	(74)代理人	100057793 弁理士 小橋 一男 (外1名)
(32)優先日	平成10年9月30日(1998.9.30)		
(33)優先権主張国	米国 (US)		

(54)【発明の名称】 記述子リング内の記述子をアップデートすることによるデータの転送制御方法及びシステム

(57)【要約】

【課題】 夫々のフレームデータバッファに関連する記述子リング及び記述子を使用することに関連するオーバーヘッドを減少させる。

【解決手段】 本発明によれば、共用システムメモリのフレームデータバッファ内にフレームが受信される。単一フレームは3個を超えるフレームデータバッファをスパンすることが可能である。記述子リングは夫々のフレームデータバッファを記述し且つそれに対してポイントし且つホスト又は装置による所有を記述する夫々の記述子を有している。フレームを受信した関連するフレームデータバッファに対する記述子が一緒に配置されて最初の記述子と最後の記述子とを有する記述子チェーンを形成する。最初の記述子及び最後の記述子と中間の記述子の所有を所望のホスト又は装置に対して許可するために最初の記述子及び最後の記述子のみが記述子チェーン内においてアップデートされる。



【特許請求の範囲】

【請求項1】 ホストと共用システムメモリを具備する制御器との間でフレームで調整されたデータ転送を制御する方法において、

単一のフレームが3個を超えるフレームデータバッファにわたるように共用システムメモリのフレームデータバッファ内のフレームを受取り、その場合に記述子リングは夫々のフレームデータバッファを記述し且つそれに対してポイントし且つホスト又は制御器のいずれかによる所有を記述する夫々の記述子を有しており、
10 最初の記述子及び最後の記述子を持った記述子チェーンを形成するために前記フレームを受取った関連するフレームデータバッファに対する記述子を一緒に配置させ、バスの利用を向上させるために所望のホスト又は制御器に対して最初の記述子及び最後の記述子及び中間の記述子の所有を許可するために前記記述子チェーン内の最初の記述子及び最後の記述子のみをアップデートする、上記各ステップを有することを特徴とする方法。

【請求項2】 請求項1において、更に、ネットワーク受信フレームに対する記述子内の所有ビットをアップデートすることによって記述子の所有をアップデートする
20 ステップを有していることを特徴とする方法。

【請求項3】 請求項1において、更に、前記記述子チェーンの最後の記述子内のフレームの終りフラグをセットするステップを有していることを特徴とする方法。

【請求項4】 請求項1において、更に、パケットの終りビットをセットすることによって前記記述子チェーンの最後の記述子内のフレームの終りフラグをセットするステップを有していることを特徴とする方法。

【請求項5】 請求項1において、更に、前記記述子チェーンを形成する複数の記述子を逐次的に一体的に配置させるステップを有していることを特徴とする方法。
30

【請求項6】 請求項1において、更に、各記述子を2ワードエントリとして形成するステップを有していることを特徴とする方法。

【請求項7】 請求項1において、更に、フレームが夫々のフレームデータバッファを充填するに従いフレームデータバッファと関連する記述子を1つづつ一体的に逐次的に配置させるステップを有していることを特徴とする方法。

【請求項8】 請求項1において、更に、128バイトワードグループとして記述子を形成するステップを有していることを特徴とする方法。

【請求項9】 請求項1において、更に、約512乃至約2,048バイトに対してフレームデータバッファを形成するステップを有していることを特徴とする方法。

【請求項10】 請求項1において、更に、前記記述子リングを前記制御器内の特定のFIFOメモリに対して専用とさせるステップを有していることを特徴とする方法。

【請求項11】 ホストと共用システムメモリを具備する制御器との間でフレームに調整されたデータの転送を制御する方法において、

単一のフレームが3個を超えるフレームデータバッファをスパンするように共用システムメモリのフレームデータバッファ内のフレームを受取り、尚記述子リングは夫々のフレームデータバッファを記述し且つそれに対してポイントし且つホスト又は制御器のいずれかによる所有を記述する夫々の記述子を有しており、

10 最初の記述子及び最後の記述子を有する記述子チェーンを形成するために前記フレームを受取った関連するフレームデータバッファに対する記述子を一緒に配置させ、所望のホスト又は制御器に対して最初の記述子及び最後の記述子及び中間の記述子の所有を許可するために記述子チェーン内の最初の記述子及び最後の記述子のみをアップデートし、

異なるフレームデータバッファにわたる全体的なフレームのバイトカウントで前記記述子チェーンの最初の記述子のメッセージ寸法フィールドをアップデートし且つその関連するフレームデータバッファ内のみのフレームのデータによって占有されているバイトカウントで前記記述子チェーンの最後の記述子のメッセージ寸法フィールドをアップデートする、上記各ステップを有することを特徴とする方法。

【請求項12】 請求項11において、更に、ネットワーク受信フレームに対する記述子内の所有ビットをアップデートすることによって記述子の所有をアップデートするステップを有していることを特徴とする方法。

【請求項13】 請求項11において、更に、前記記述子チェーンの最後の記述子内のフレームの終りフラグをセットするステップを有していることを特徴とする方法。

【請求項14】 請求項11において、更に、パケットの終りビットをセットすることによって前記記述子チェーンの最後の記述子内のフレームの終りフラグをセットするステップを有していることを特徴とする方法。

【請求項15】 請求項11において、更に、前記記述子チェーンを形成する記述子を逐次的に一緒に配置させるステップを有していることを特徴とする方法。

40 【請求項16】 請求項11において、更に、各記述子を2ワードエントリとして形成するステップを有していることを特徴とする方法。

【請求項17】 請求項11において、更に、フレームが夫々のフレームデータバッファを充填するに従いフレームデータバッファと関連する記述子を1つづつ一緒に逐次的に配置させるステップを有していることを特徴とする方法。

50 【請求項18】 請求項11において、更に、記述子を128バイトワードグループとして形成するステップを有していることを特徴とする方法。

【請求項 19】 請求項 11 において、更に、約 512 乃至約 2,048 バイトに対してフレームデータバッファを形成するステップを有していることを特徴とする方法。

【請求項 20】 請求項 11 において、更に、前記制御器内の特定の FIFO メモリに対し記述子リングを専用とさせるステップを有していることを特徴とする方法。

【請求項 21】 ホストと共用システムメモリを具備する制御器との間でフレームに調整されたデータの転送を制御する方法において、
10 最初の記述子と最後の記述子とを有する記述子チェーンを形成するためにフレームを受信した関連するフレームデータバッファに対してポイントする複数の記述子を一緒に配置させ、
バスの利用を向上させるために所望のホスト又は制御器に対して最初の記述子及び最後の記述子及び中間の記述子の所有を許可するために記述子チェーン内の最初の記述子及び最後の記述子のみをアップデートする、上記各ステップを有することを特徴とする方法。

【請求項 22】 請求項 21 において、更に、異なるフ
20 レームデータバッファにわたる全体的なフレームのバイトカウントで前記記述子チェーンの最初の記述子のメッセージ寸法フィールドをアップデートするステップを有していることを特徴とする方法。

【請求項 23】 請求項 22 において、更に、それと関連するフレームデータバッファ内のみに含まれているフレームのデータによって占有されているバイトカウントで前記記述子チェーンの最後の記述子のメッセージ寸法フィールドをアップデートするステップを有していることを特徴とする方法。

【請求項 24】 請求項 21 において、更に、ネットワーク受信フレームに対する記述子内の所有ビットをアップデートすることによって記述子の所有をアップデートするステップを有していることを特徴とする方法。

【請求項 25】 請求項 21 において、更に、前記記述子チェーンの最後の記述子内のフレームの終りフラグをセットするステップを有していることを特徴とする方法。

【請求項 26】 パケットの終りビットをセットすることによって前記記述子チェーンの最後の記述子内のフ
40 レームの終りフラグをセットするステップを有していることを特徴とする方法。

【請求項 27】 請求項 21 において、更に、前記記述子チェーンを形成する複数の記述子を逐次的に一緒に配置させるステップを有していることを特徴とする方法。

【請求項 28】 請求項 21 において、更に、各記述子を 2 ワードエントリとして形成するステップを有していることを特徴とする方法。

【請求項 29】 請求項 21 において、更に、フレーム 50

が夫々のフレームデータバッファを充填するに従いフレームデータバッファと関連する記述子を 1 つづつ一緒に逐次的に配置させるステップを有していることを特徴とする方法。

【請求項 30】 請求項 21 において、更に、前記記述子を 128 バイトワードグループとして形成するステップを有していることを特徴とする方法。

【請求項 31】 請求項 21 において、更に、約 512 乃至約 2,048 バイトに対してフレームデータバッファを形成するステップを有していることを特徴とする方法。

【請求項 32】 請求項 21 において、更に、前記制御器内の特定の FIFO メモリに対して前記記述子リングを専用のものとするステップを有していることを特徴とする方法。

【請求項 33】 フレームに調整されたデータの転送を制御するシステムにおいて、

ホストシステム、

ネットワーク装置、

前記ホストシステムと前記ネットワーク装置との間の共用システムメモリであってフレームデータバッファを具備している共用システムメモリ、

単一のフレームが 3 個を超えるバッファにわたることが可能であるように前記フレームデータバッファ内に入って来るフレームを受信する手段、

フレームを受信した夫々のフレームデータバッファに対してポイントする記述子チェーンに形成した連続する記述子を具備する記述子リングを前記メモリ内に形成する手段、

30 バスの利用を向上させるために所望のホスト又はネットワーク装置に対して最初の記述子及び最後の記述子及び中間の記述子の所有を許可するために前記記述子チェーン内の最初の記述子及び最後の記述子のみをアップデートする手段、を有していることを特徴とするシステム。

【請求項 34】 請求項 33 において、前記記述子が、更に、アップデートされる所有ビットを有していることを特徴とするシステム。

【請求項 35】 請求項 34 において、各記述子が、更に、2 ワードエントリを有していることを特徴とするシステム。

【請求項 36】 請求項 33 において、前記記述子が約 128 バイトワードグループを有していることを特徴とするシステム。

【請求項 37】 請求項 33 において、前記フレームデータバッファが約 512 乃至約 2,048 バイトを有していることを特徴とするシステム。

【請求項 38】 フレームに調整されたデータの転送を制御するシステムにおいて、

ホストシステム、

受信ポートと送信ポートとを具備しており各ポートが F

I F O メモリを有している制御器、
前記ホストシステムと前記制御器との間の共用システム
メモリであってフレームデータバッファを具備している
共用システムメモリ、
単一のフレームが 3 個を超えるバッファにわたることが
可能であるように前記フレームデータバッファ内に入っ
て来るフレームを受信する手段、
フレームを受信した夫々のフレームデータバッファに対
してポイントする記述子チェーンに形成した連続する記
述子を有する少なくとも 1 個の記述子リングであって各
々が前記制御器内の特定のポート及び F I F O メモリに
10 に対して専用とされている記述子リングを前記メモリ内
に形成する手段、
バスの利用を向上させるために所望のホスト又は制御器
に対して最初の記述子及び最後の記述子及び中間の記述
子の所有を許可するために前記記述子チェーン内の最初
の記述子及び最後の記述子のみをアップデートする手
段、を有していることを特徴とするシステム。

【請求項 3 9】 請求項 3 8 において、前記記述子がア
ップデートされる所有ビットを有していることを特徴と
20 するシステム。

【請求項 4 0】 請求項 3 8 において、各記述子が 2 ワ
ードエントリを有していることを特徴とするシステム。

【請求項 4 1】 請求項 3 8 において、前記記述子が約
1 2 8 バイトワードグループを有していることを特徴と
するシステム。

【請求項 4 2】 請求項 3 8 において、前記フレームデ
ータバッファが約 5 1 2 乃至約 2 , 0 4 8 個のバイトを
有していることを特徴とするシステム。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】 本発明は、ホストと共用シ
ステムメモリを具備する制御器との間でフレームに調整さ
れたネットワークをベースとしたデータの転送を制御す
る方法に関するものであって、更に詳細には、フレーム
データバッファを記述し且つそれに対してポイント即ち
指し示す夫々の記述子を具備する記述子リングを使用し
てデータの転送を制御する方法及びシステムに関するも
のである。

【0 0 0 2】

【従来の技術】 データネットワークは毎日の活動及びビ
ジネスアプリケーションにおいて益々重要なものとなっ
ている。これらのネットワークの殆どは、例えば、しば
しば T C P / I P として呼称される伝送制御プロトコル
(T C P) 及びインターネットプロトコル (I P) を使
用するインターネットのようなパケット交換型ネットワ
ークである。伝送制御プロトコルはネットワークトラ
フィックの信頼性のある受信及び送信を管理し、一方イン
ターネットプロトコルはパケットが正しいデスティネ
ーション即ち宛先へ送られることを確保するためのルーチ
50

ング即ち経路付けの役割を担っている。

【0 0 0 3】 典型的なネットワークにおいては、スイッ
チングノード及び端部ノードのみならず伝送リンクから
なるメッシュが設けられる。端部ノードは、典型的に、
パケットが受信され且つそのデスティネーション即ち宛
先へ到達するために正しい外へ出るリンク上に送信され
ることを確保する。スイッチングノードは、典型的に、
パケットスイッチ、又はルータ、又は中間システムとし
て呼称される。データトラフィックにおけるソース即ち
発信元及びデスティネーション即ち宛先(端部ノード)
はホスト及び端部システムとして呼称することが可能で
ある。これらのホスト及び端部システムは、典型的に、
パソコン、ワークステーション及びその他のターミナル
である。

【0 0 0 4】 コンピュータの間においての情報の移動を
助けるために、開放型システム間相互接続 (O S I) モ
デルが開発されている。コンピュータの間での情報を移
動させる各問題は該モデルにおいてはレイヤ(層)によ
って表わされ、従って、標準用の枠組を確立している。
2 つのシステムはプロトコルスタック内のレイヤの間に
おいてのみ通信を行う。然しながら、他のシステム内の
純粋なレイヤと通信を行うことが望ましく、且つこのよ
うな結果を達成するために、情報はパケットとしても知
られるプロトコルデータ単位 (P D U) によって交換さ
れる。P D U はデータのみならずアドレスのような制御
情報を包含するヘッダを有している。ソース即ち発信元
において、当業者にとって公知の如く、各レイヤはそれ
自身のヘッダを付加する。物理レイヤから始まって、7
個のレイヤ、即ち (1) 物理レイヤ、(2) データリン
クレイヤ、(3) ネットワークレイヤ、(4) トランス
ポートレイヤ、(5) セッションレイヤ、(6) プレゼ
ンテーションレイヤ、(7) アプリケーションレイヤを
有している。

【0 0 0 5】 ネットワークシステムは、典型的に、ルー
チングアルゴリズムを使用することによって最適な経路
を決定することが可能なルータを使用する。ルータは、
又、各パケットに対するルーチング経路に基づいて、入
力ポートに到着したパケットを出力ポートへスイッチさ
せる。ルーチングアルゴリズム(即ちルーチングプロト
コル)は、与えられたデスティネーションアドレスを具
備するパケットを送るために次のルータへポイントする
即ち指し示すエントリからなるルーチングテーブルを初
期化し且つ維持するために使用される。典型的に、ネッ
トワーク内の各リンクに対して固定したコストが割り当
てられ且つコストはリンク帯域幅及び/又はコストを反
映する。最も低いコストの経路が他のルータとネットワ
ークトポロジ及びリンクコスト情報を交換した後ル
ータによって決定することが可能である。

【0 0 0 6】 2 つの低いレベルのレイヤ、即ち物理レイ
ヤ及びデータリンクレイヤは、典型的に、I E E E 8 0

2 委員会によって開発されたローカルエリアネットワーク用の標準によって支配される。データリンクレイヤは、典型的に、2つのサブレイヤに分割されており、論理リンク制御（LLC）サブレイヤは例えばフレーム化、フロー制御、エラー制御、アドレッシング等の機能を定義する。LLCプロトコルはHDLCプロトコルの修正版である。媒体アクセス制御（MAC）サブレイヤは共通媒体に対する伝送アクセスを制御する。

【0007】ハイレベルデータリンク制御（HDLC）は遠隔装置間でのデータ伝送操作の正確性をチェックするための通信制御手順であり、その場合には、データはフレームとして知られる単位で伝送され、且つフレームのシーケンスをチェックし且つ伝送操作期間中にビットが失われるか又は反転されることに起因したエラーを検知するための手順が存在している。データリンクのセットアップ（設定）及び終了を制御する機能も存在している。HDLCにおいては、伝送リンクにわたってのビット同期データ通信が制御される。HDLCはX. 25として知られているITUパケット交換インターフェース標準に包含されている。

【0008】プログラム可能なHDLCプロトコル制御器はこれらのシステムにおいて一般的に使用されている。HDLC制御器はコンピュータペリフェラルインターフェース装置であって、それは国際標準化機構（ISO）ハイレベルデータリンク制御（HDLC）をサポートしている。それはフレームレベルの命令セットをサポートすることにより且つフレーム組立・分解及びデータ完全性に関連する低レベルタスクのハードウェア実現によって中央処理装置又はマイクロプロセッサユニット（MPU）ソフトウェアを減少している。

【0009】殆どの通信プロトコルはビット指向型で、コード依存性であり、且つ全二重通信に対して理想的なものである。幾つかの通信適用例としては、ターミナル対ターミナル、ターミナル対MPU、MPU対MPU、衛星通信、パケット交換、及びその他の高速データリンク等がある。

【0010】通信制御器はフレームを構成し且つ受信することに関連するタスクの多くから中央MPUを解放する。フレーム（時折、パケットと呼称される）はリンク制御及びデータ伝送の両方の目的に使用することの可能な単一の通信要素である。

【0011】殆どの制御器は外部共用メモリ資源に対するアクセスを与える直接メモリアccess（DMA）装置又は機能を有している。その制御器はDMA又は非DMAのいずれかのデータ転送を行うことを可能とする。該制御器は、MPUからのコマンドを受け、そのコマンドを実行し、且つインタラプト及び結果をMPUへ供給する。

【0012】幾つかのHDLC制御器は半又は全二重通信をサポートする異なるインターフェース（例えば、1

0Mbps又は高々100Mbps）を有する種々のポートを有している。フレームのうちの幾つかはバス内へ移動し、次いで、直接メモリアccess（DMA）ユニット内へ移動し、そこでCPU及びROMは専用の命令セットを有するファームウェアによって制御される。パケットがポートに入るとインタラプトが発生し、且つCPUがインタラプト即ち中断される。ファームウェアコードはDMAに対するコマンドを「駆動」し且つパケットは共用外部バスから外に出てメモリ内に書込まれる。

【0013】例えばHDLC制御器及びその他のフレームを受信し且つ送信する同様の装置等の任意のデータ構造は、フレームを受信するためにそれらを実現することが可能である多数の方法を有している。幾つは例えばリンクリスト及びリングバッファ等のダイナミックメモリ割当て手法を有している。別の実現例は、リンクリストを使用するものであり、一方その他のものはリングバッファを使用するものであり、例えば、マーケットランス（Market Lance）MK 5032として知られており、後に、AMD7990チップと名前が変えられた最初のイーサネットチップで実現された最初のタイプのリングバッファアルゴリズム等がある。

【0014】典型的に、リングバッファにおいては、ホストと制御器との間の共用システムメモリ内の夫々のフレームデータバッファを記述し且つそれに対してポイント即ち指し示す夫々の記述子を有するディスクリプタリング即ち記述子リングが存在している。該記述子リングはポイント及びフレームデータバッファに対する情報を有する記述子エントリを具備する循環型キュー（待ち行列）である。各記述子リングはHDLC制御器内の特定のFIFOメモリに対して専用である。記述子リング内の各2ワード記述子エントリはシステムメモリ内の1つの特定のバッファと関連しており、該システムメモリは、例えば、制御器及びホスト等のネットワーク装置間の共用システムメモリである。

【0015】フレームデータバッファは、典型的に、送信用のフレームを包含するか又はフレーム受信用の空間を提供するメモリブロック（典型的に、512乃至2,048バイトの範囲）として定義される。当然、各送信チャンネル及び各受信チャンネルは専用の記述子リングを使用する。フレームが単一フレームデータバッファの有限の容量を超える場合には、そのフレームは該バッファを「スパン」する、即ちそのバッファにわたっていると云われる。各記述子の最初のワードにおける所有ビットは、ホスト又は制御器が関連するフレームデータバッファを所有しているか否かを表わす。

【0016】所有性は制御器及びホストによって準拠されねばならない特定のプロトコルに従う。記述子の所有性が他の装置又はホストに対して放棄され、且つ記述子の一部とされると、その関連するフレームデータバッファを変更することが可能である。ホストはフレーム受

10

20

30

40

50

信用の空のフレームデータバッファ及びフレーム送信用の満杯のフレームデータバッファの所有をネットワーク装置に与える。逆に、ネットワーク装置はそれが使用した送信フレームデータバッファ及びそれが充填した受信フレームデータバッファに対する所有をホストへパスする。

【0017】フレーム受信の場合には、ホストは空のフレームデータバッファに対してポイントする即ち指し示している連続する記述子の所有を制御器又はその他のネットワーク装置へ与えることが要求される。フレームが制御器によって完全に受取られると、その構成要素の記述子の所有は再度割当てられる。ホストはインタラプトを介してそのイベントに関する信号が与えられる。ホストは、典型的に、その信号の意味を推測するためにレジスタを読取ることの義務が課される。このことが達成されると、そのフレームは、何等かの態様でディスパッチ即ち送給され且つ関連性のある記述子の所有は制御器へリターンされる。

【0018】典型的な動作においては、ホストは制御器又はリングの周りのその他のネットワーク装置に「追従」し、制御器が使用するためにその後「空の」記述子を残す。該装置がホストの余りにも前に行き過ぎると、それは記述子リングをラップアラウンドしてそれが所有することのない記述子に遭遇する場合がある。その結果、このことが発生すると、入って来るフレームが失われる場合がある。フレーム送信の場合には、該装置は送信記述子リングの周りをホストを「追従」し、ホストが獲得するためにその後使用済の記述子を残す。ホストは、それが送信の準備がなされている1つ又はそれ以上のフレームを有する場合にデバイスに対して記述子の所有を与えるに過ぎない。フレームが該装置によって完全に送信されると、その構成要素の記述子の所有は再使用のためにホストへパスされる。ホストは、インタラプトを介してこのイベントについて信号が与えられる。

【0019】幾つかの適用例においては、ホストが送信器として受取られるフレームよりも寸法が小さいフレームデータバッファを使用することを選択することが可能である。換言すると、単一のフレームが複数のバッファをスパンする、即ち複数のバッファにわたる場合がある。このタイプのシステムは、制御器によってフレームを切断する（受信時に散乱させる）か又は組立てる（送信時に寄せ集める）ことを可能とする。複数のフレームデータバッファは、関連する記述子を一緒に「チェーン化」即ちグループ化することによってフレームの構成要素の断片を保持することが可能である。定義上、一緒にチェーン化又はグループ化された記述子は記述子リングにおける連続的なエントリであり、チェーンの終端記述子においてフレームの終りフラグがセットされる。所有されているがそのフレームの終りフラグ（EOF）がセットされていない記述子エントリのフレームデ

ータバッファは全体的なフレームではなくフレームの一部であると考えられる。

【0020】大きなフレームの受信期間中に、該装置はそれが各相次ぐフレームデータバッファを充填するに従い、1つづつ記述子を一緒に「チェーン化」即ちグループ化する。フレームの終りが受取られ且つ共用システムメモリへ転送されると、フレームの終りフラグが記述子チェーンの決定用記述子内にセットされる。送信期間中に、制御器はチェーン化されたバッファの内容から単一のフレームを逐次的に構築することが可能である。記述子がフレームの終りフラグをセットしているバッファに遭遇する場合にのみ、フレームの送信が終了する。

【0021】公知のシステムにおいては、各スパンされた記述子は連続的にアップデートされ且つ変更されていた。従って、充分なるバスCPU資源が割当てられねばならなかった。このことはホストが達成するのに高価な動作である。コマンドが構築され且つDMAへ供給されねばならず且つDMAはファームウェアでバスを再調停せねばならない場合がある。システムバスは調停されねばならない場合があり、一方その他の装置もシステムバスへアクセスしようとする場合がある。記述子チェーン内に全ての記述子を書き戻すことは著しい量の資源を使用する。

【0022】

【発明が解決しようとする課題】本発明は、以上の点に鑑みなされたものであって、上述した如き従来技術の欠点を解消し、夫々のフレームデータバッファと関連する記述子及び記述子リングの使用に関連するオーバーヘッドを減少させることを目的とする。

【0023】

【課題を解決するための手段】本発明によれば、フレームが3個又はそれ以上のフレームデータバッファ等のフレームデータバッファを「スパン」する場合、即ちフレームデータバッファにわたっている場合に、バスの利用が向上される。本発明方法及び装置によれば、最初の記述子及び最後の記述子のみがアップデート即ち更新される。本装置がチェーン化されたフレーム内に関与しているフレームデータバッファを終了すると、それは、最初に、最後の記述子の所有をリターンし、次いで、それは最初の記述子の所有をリターンし、それはフレームの「フェンスポスト（fence post）」即ち「垣根の支柱」を形成するものである。ホストは、例えばそれらが該装置によって所有されているものであっても、全ての中間に介在するフレームデータバッファの所有を獲得する。従って、ホストがフレームの終りフラグ（EOF）によってマークが付けられていないホストによって所有されている記述子に遭遇すると、フレームの終りフラグがセットされている次のホストによって所有されている記述子に至るまで且つそれを包含して、全ての連続の記述子及び関連するフレームデータバッファの所有を

獲得する。

【0024】本発明によれば、ホストと共用システムメモリを具備するネットワーク装置との間においてフレームに調整されているデータの転送を制御する方法が、単一のフレームが3個を超えるフレームデータバッファをスパンするように即ち3個を超えるフレームデータバッファにわたるように共用システムメモリのフレームデータバッファ内のフレームを受取るステップを有している。記述子リングは、夫々のフレームデータバッファを記述すると共にそれに対してポイントし且つホスト又は装置のいずれかによる所有を記述する夫々の記述子を有している。フレームを受取った関連するフレームデータバッファに対する記述子は、一緒に配置されて最初の記述子と最後の記述子とを有する記述子チェーンを形成する。最初の記述子及び最後の記述子のみが記述子チェーン内においてアップデートされ、所望のホスト又は制御器に対して最初の記述子及び最後の記述子及び中間の記述子の所有を許可してバスの利用を向上させる。

【0025】本方法は、更に、ネットワーク受信フレームに対する記述子内の所有ビットをアップデートすることによって記述子の所有をアップデートするステップを有している。本方法は、更に、記述子チェーンの最後の記述子内のフレームの終りフラグをセットするステップを有している。本方法は、更に、パケットの終りビットをセットすることによって記述子チェーンの最後の記述子内のフレームの終りフラグをセットするステップを有している。本方法は、更に、記述子チェーンを形成する記述子を逐次的に一緒に配置させるステップを有している。各記述子は2ワードエントリとして形成することが可能であり且つフレームデータバッファと関連する記述子はフレームが夫々のフレームデータバッファを充填するに従い1つずつ逐次的に一緒に配置させることが可能である。記述子は、又、128バイトワードグループとして形成することも可能である。フレームデータバッファは約512乃至2,048バイトであるように形成することが可能である。記述子リングはネットワーク装置内の特定のFIFOメモリに対して専用のものとする

ことが可能である。

【0026】本発明によれば、システム及び関連する装置がフレームに調整されているデータの転送を制御し且つホストシステムとネットワーク装置とを包含している。共用システムメモリがホストシステムと装置との間に存在している。共用システムメモリはフレームデータバッファを有している。単一フレームが3個を超えるバッファをスパンすることが可能であるようにフレームデータバッファ内に入って来るフレームを受取る手段が設けられている。フレームを受取った夫々のフレームデータバッファに対してポイントする、即ち指示する記述子チェーンに形成された隣接する記述子を有する記述子リングをメモリ内に形成する手段が設けられている。バス

の利用を向上させるために所望のホスト又は装置に対して最初及び最後の記述子及び中間の記述子の所有を許可するために記述子チェーン内の最初の記述子及び最後の記述子のみをアップデートする手段が設けられている。

【0027】装置は、又、フレームに調整されたデータの転送を制御することが可能であり、且つホストシステム及び装置のみならず、受信ポート及び送信ポートを具備する装置も包含している。各ポートはFIFOメモリを有している。共用システムメモリがホストシステムと装置との間に存在している。共用システムメモリはフレームデータバッファを有している。単一のフレームが3個を超えるバッファをスパンすることが可能であるようにフレームデータバッファ内に入って来るフレームを受信する手段が設けられている。フレームを受取った夫々のフレームデータバッファに対してポイントする記述子チェーン内に形成されている隣接する記述子を有する少なくとも1個の記述子リングをメモリ内に形成する手段が設けられている。これらの記述子リングは装置内の特定のポート及びFIFOメモリに対して専用のものとされる。バスの利用を向上させるために、所望のホスト又は制御器に対して最初の記述子及び最後の記述子及び中間の記述子の所有を許可するために記述子チェーン内の最初の記述子及び最後の記述子のみをアップデートする手段が設けられている。

【0028】

【発明の実施の態様】本発明を本発明の好適実施例が示されている添付の図面を参照して詳細に説明する。然しながら、本発明は、多くの異なる態様で実現することが可能であり且つ以下に説明する実施例にのみ制限されるべきものとして解釈されるべきではない。そうではなく、これらの実施例は本発明の開示が完全なものであり且つ当業者にとって本発明の範囲を完全に伝達するように提供されるものである。尚、本明細書にわたって同一の構成要素には同一の参照番号を使用している。

【0029】図1-3を参照し、特に図1及び2を参照すると、本発明の1例であるネットワーク制御器とホストシステムのハイレベルの概略図が示されている。ネットワーク制御器は本発明の1つの特定の実施例においてはHDL C制御器である。

【0030】本発明はネットワーク制御器を利用する従来のネットワークを包含する多数の異なるネットワークにおいて使用することが可能である。例えば、本発明はインターフェースカードからインターフェースカードへ延在するケーブルによってコンピュータが接続されている多くのローカルエリアネットワークにおいて使用することが可能である。配線用のハブは各ネットワークインターフェースカードへ取付けられているケーブルに対する中央点を提供することが可能である。ハブは例えば同軸、オプティカルファイバ、ツイスト対ワイヤ等のコネクタを接続することが可能である。1つのタイプの形態

は、10ベースTとして知られる非シールド型ツイスト対ワイヤを使用することが可能である。何故ならば、それは、毎秒10メガビット(MBPS)の信号速度、直流即ちベースバンドの信号処理、及びツイスト対ワイヤを使用するものだからである。

【0031】ネットワークは、典型的に、ネットワーク(Net Ware)IPXプロトコルに含まれているデスティネーション(宛先)アドレスを検査するもの等のルータを有することが可能である。ルータはインターネットパケット、リングフレーム又はその他の情報を剥取り且つIPXパケット及びそのカプセル化したデータをリンクを横断して送ることが可能である。ブリッジは各インターネットパケットのアドレスを検査し且つそれを回路を横断して送ることが可能である。

【0032】図1は典型的なハイレベルシステムの概略図を示しており、それは本発明の一般的な方法、装置及びシステムの例示である。図示されるように、ネットワーク装置としても知られている4個のネットワーク制御器40がホストシステム43へ接続している32ビットシステムバスへ接続している。ホストマイクロプロセッサ44が、共用メモリサブシステム46と同様にシステムバス42へ接続している。各制御器40は4個のポート50、52、54、56を有しており、それらは夫々のハイレベルデータリンク制御レイヤである全二重プロトコル線58へ接続している。

【0033】各ネットワーク制御器40は次世代のブリッジ及びルータ装置、及びT3速度におけるHDL動作を必要とする装置において使用するように設計されている高性能の4ポート高速ネットワーク制御器である。各ネットワーク制御器は、好適には、単一のチップとして製造される。

【0034】図2に示したように、ネットワーク側においては、ネットワーク制御器40が前述した如く且つ0乃至3の番号が付けられた4個のポート50、52、54、56を有しており、その各々は別個の送信及び受信FIFOを具備しており、半二重又は全二重動作を可能としている。各ポート50-56は送信データハンドラ60を有しており、それは送信クロック信号(TCLK)を受取り且つデータ信号(TData)をライントランシーバ62へ転送する。受信データハンドラ64もクロック信号(RCLK)を受取り且つデータをライントランシーバ62へ送り且つそれから受取る。該ポートは、又、各々、図示した送信及び受信先入先出(FIFO)論理回路66、68、512バイト送信FIFO70、制御回路74、512バイト受信FIFO72を有している。512バイトFIFO70、72はフレームバス76へ接続しており且つ制御回路74は管理バス78へ接続している。FIFO論理回路66、68及びデータハンドラ60、64及び制御回路74は送信及び受信(Tx)、(Rx)512バイトFIFO用の適宜

の送信及び受信回路として動作する。

【0035】システム側においては、制御器40は高速(25乃至33MHz)の32ビットシステムバスインターフェース制御ユニット(SBI)80を有しており、それは制御器のシステムバスの使用を最小とさせ且つその性能を最大とさせるために単一サイクルワード転送を使用する。直接メモリアクセスユニット(DMA)動作は該装置がバスマスタとなることを可能とし、且つ格納及び転送アプリケーションに対する効率的なバッファ管理アルゴリズムを使用することが可能である。システムバスインターフェース制御ユニット80は共用バスインターフェース回路82、バススレーブ制御器84、DMA制御器でもあるDMAバスマスタ制御器即ち直接メモリアクセスユニット85、コンフィギュレーション(形態特定)データ転送エンジン86、管理データ転送エンジン88(それは両方とも管理バス78に対して通信を行う)、且つフレームバス76に対して通信を行うフレームデータ転送エンジン90を有している。

【0036】ユーザによって直接的にアクセス可能なものではないが、ネットワーク制御器は、又、通信プロセッサコア又は単に通信プロセッサ(CPC)92と呼ばれる埋込型の32ビットRISCプロセッサを有している。CPCは、ポート毎の統計の収集、DMAモードバッファ管理及びデータ転送、チップ自己テスト及びホスト/チッププリミティブコマンド/応答交換等の活動を取扱う。CPC92はCPU94、ALU96、タイマ98、RAM100、ファームウェアROM102、インタラプトハンドラ104を有している。

【0037】管理及びフレームデータ転送を効率的な態様でサポートするために、内部バスが制御器のサブシステムの全てを接続している。別のバス、及び管理バス78及びフレームバス76は夫々の管理データ及びフレームデータに対して使用され、並列性を増加させ且つそれにより性能を増加させる。制御器40は当業者によって公知の方法によってチップ上に形成される。

【0038】格納及び転送適用例に対して設計される場合には、ネットワーク制御器40は32ビットデータ又はフレームバス42を介してシステムメモリと8個のオンチップの512バイトFIFO70、74との間でフレームを転送するためにオンチップのDMAエンジン及び効率的なバッファ管理アルゴリズムを使用する。この動作においては、制御器40はネゴシエーションによってバスマスタとなり、システムバスの所有を獲得し、次いで、チップとシステムメモリ46との間でフレーム及び管理データを直接的に移動させる。ホストプロセッサ44はバススレーブモードで動作する同一のバスを使用することによって制御器のオンチップのコンフィギュレーション(形態特定)/ステータスレジスタへ直接アクセスすることが可能である。

【0039】通信プロセッサ92は別個のプログラム及び

データベースを具備するハーバード (Harvard) 型アーキテクチャを使用し、該バスは同時的なデータのトランザクションをサポートする。典型的に、クロックサイクル当たり1個の命令を効果的に実行するために4ステージパイプライン型制御ユニットが使用されている。このアーキテクチャによって必要とされる高性能を与えるために、通信プロセサによって使用されている内部SRAM100は3個のポートを有することが可能であり、且つ、典型的に、トライポートRAM (TPR) と呼称される。このアーキテクチャを使用することにより、1個のレジスタ (TPR) からの読取、ALU演算、異なるレジスタ又はTPR位置への書込の全てが1つの命令での同一のクロックサイクル内において発生することを可能とする。

【0040】制御器の動作 (バッファ管理及びデータ転送、チップ自己テスト及びホスト/チッププリミティブコマンド/応答交換、及び統計収集を包含する) を制御するファームウェアプログラムはROM102内に包含されており、それはオンチップの8KROMとすることが可能である。

【0041】ネットワーク制御器40は外部的に供給されたシステムクロックから内部システムクロックを発生するためにフェーズロックループ (PLL) を使用している。このPLLによって発生されたシステムクロックは性能に影響を与える場合のある信号対システムクロックチェーンを最小とするように遅延される。その結果、制御器システムクロックは25又は33MHzでなければならない。

【0042】説明の便宜上、図1-8を参照して動作の外観について説明し、次いでその後の図面を参照して動作のより詳細について説明する。制御器が初期化され且つポートがアップし且つ稼動状態となると、典型的なフレーム受信が以下の如くに進行する。フレームの開始フラグの二進01111110パターンがHDLポート受信器回路によって検知され、それはRx FIFO論理68、Rxデータハンドラ64、ライントランシーバ62を有している。この直列のデジタルデータストリームはHDLポートの受信器回路へ流れ、そこでフレームの始め (非フラグパターン) に対するサーチが行われてオクテット (octet) の整合及びフレームの始めが確立される。フレームチェックシーケンス (FCS) の計算が実際のフレームの後の最初のオクテットに関して開始される。

【0043】直列対32ビット並列ワード変換が該受信器回路によって実施され且つデータワードが受信器 (Rx) FIFO74内に格納される。この動作の始めにおいてRx FIFO74が空であったと仮定すると、受信データは受信FIFO74内のワード数がプログラムされているウォーターマーク (watermark) 設定よりも大きくなるまで、継続して受信FIFO74を充

填する。以下に更に詳細に説明するように、この点において、オンチップRISC92上で稼動しているファームウェア102に対して受信FIFO74に対するデータ転送を要求するインタラプトが発行される。このインタラプトはネットワーク制御器42内部のものであってホストシステム44からは見えないものである。

【0044】インタラプトを受取ると、ファームウェア102は要求するポートに対する現在の受信記述子 (前にフェッチしたもの) のオンチップコピーをチェックする。それがバッファの所有を有するものでない場合には、オンチップDMAが検査のために適切な記述子を再度フェッチすべく指示する。制御器40は2つのイベントのうちの1つが発生するまで繰返し記述子をフェッチする。即ち、(1) バッファの所有がそれに与えられるか、又は(2) 受信FIFOがオーバフロー (この場合にはフレームが失われる) のいずれかである。バッファの所有が許可されると、ファームウェアは、受信 (Rx) FIFO74からシステムメモリ内の受信バッファへバースト寸法のフレームデータワードを転送すべくDMAに指示することによってこのインタラプトに応答する。第一バーストの受信フレームをシステムメモリへ送信すると、マスタインタラプトレジスタ (MIR) を介してホストに対してFAN (フレームアドレス通知) インタラプトを発生することが可能である。

【0045】受信FIFO74充填 (ネットワーク制御器受信器回路による)、受信器対ファームウェアのインタラプト、FIFOを空にすること (DMA) によるサイクルが、フレームの終りが受信器回路によって遭遇されるまで継続して行われる。この点において、該フレームのフレームチェックシーケンス (FCS) が受信器回路によってチェックされ且つ受信ステータスワードが発生され且つ受信FIFO74内のフレームの後にアペンド即ち添付される。以下に説明するように、フレームの残り及び受信ステータスワードがシステムメモリ内の受信バッファへ転送されるまで受信器対ファームウェアインタラプトは継続する。ファームウェアは受信記述子内の所有、メッセージ寸法、エラーフラグ等をアップデートするためにオンチップDMA85を使用し、次いで、完了した受信を表わすマスタインタラプトレジスタ (MIR) (図8B) を介してホストに対して「フレーム受信済」インタラプト (RINT) を発行する。

【0046】典型的なフレーム送信は以下のようにして行われる。全てのフレームは送信記述子リング202

(図3) におけるエントリに対して割当てられている送信フレームデータバッファ204からネットワーク制御器40によって送信される。システムがフレームを送信するためにネットワーク制御器40に対して準備がされている場合には、それは関連する送信記述子の所有を放棄し、次いで、2つのことのうちの1つを行う。即ち、

(1) 制御器の送信ポールタイマが期間満了し、チップ

をしてそれが所有するバッファを探すためにTx記述子をポーリングすることを待機するか、又は(2)ホストによるシステムモードレジスタ(SMR)を介しての送信要求(TDMD)が発行されるかのいずれかである。いずれの場合においても、ファームウェアはバッファからバースト寸法の量のフレームデータをフェッチし且つそれを適宜のポートの送信FIFO内に配置させることを開始すべくDMAに命令する。これは、FIFOがプログラムされているウォーターマークより上に充填されるまで又はフレームの終りに到達するまで継続する。

【0047】プログラムされている送信開始点を満足するのに十分なワードが送信FIFO70内に存在すると、送信データハンドラ60、送信FIFO論理66、ライントランシーバ62を包含する送信器回路が送信を開始する。この送信器回路は並列対直列変換を行い、連続する直列データストリームを送り出す。開始フラグが送られ、それに続いてフレームデータ及びそのフレームに対するサイクル冗長性チェック(CRC)又はFCSが送られる。フレームチェックシーケンス(FCS)の計算はフレームの最初のオクテットで開始する。送信FIFO70がウォーターマーク設定より低い空の状態になると、送信器回路はオンチップファームウェア102に対してプライベートインタラプトを発行し、システムメモリから更なるデータをコピーすることを要求する。

【0048】空にすること(送信器ユニットによる)及び充填すること(DMAによる)のサイクルは、フレームの終り(EOF)が該FIFO内に書き込まれるまで継続して行われる。送信器が送信FIFOからそのフレームの最後のデータを除去すると、それは、オプションとして、それが計算したFCSをアペンド即ち添付する(制御器によるFCSのアペンド処理はフレーム毎に制御することが可能である)。送信器はクロージングフラグ即ち閉じるためのフラグを送ることによってフレームを閉じる。

【0049】ネットワーク制御器40内部の埋込型プロセッサ92は、ホストシステムが使用するためにオンチップレジスタ内に12個の統計を維持する。これらの統計はバススレーブコンフィギュレーション/ステータスレジスタ動作を使用してホストによってアクセスされる。付加的な特徴として、該制御器は以下に説明するようにシステムメモリ内のオンチップ統計の完全なコピーを配置させるためにそのオンチップDMAを使用することが要求される場合がある。

【0050】システムバスインターフェースユニット(SBI)80はDMAモードにおいて3つの重要な機能を実施する。即ち、(1)HDLCフレームデータ転送用のDMAエンジン(バスマスタ)、(2)コンフィギュレーション/ステータスレジスタへアクセスするためのマイクロプロセッサポート(バススレーブ)、(3)効果的には2個のインタラプトピン(MINTR#及び

PEINTER#)に対するソース即ち発信元である。バスマスタ及びバススレーブの両方の動作は同一の32ビットデータバスを使用し且つ同一の制御信号の幾つかを共用する。バススレーブ動作(CBIG)及びバスマスタ動作(TBIG)に対する適切なモードを選択するために別個のピンが存在している。

【0051】システムバスインターフェースユニット(SBI)80はホストプロセッサ44の関与なしで共用バス42を介してシステムメモリ46とのブロックデータ転送を実施するためのマルチチャンネルDMAユニット85を有している。該制御器は、それが管理ブロック200、送信又は受信記述子206、又は送信又は受信フレームデータバッファ204へのアクセスを必要とする場合にシステムバスの所有を要求し、尚、それについては図3を参照して後に説明する。

【0052】ネットワーク制御器40がこれらのデータ構成体のうちの1つへアクセスする毎に、それはバスの所有に関してネゴシエーションを行い、データ(これは幾つかのワードである場合がある)を転送し、次いで、バスの所有を放棄する。与えられたバスの所有に対して、逐次的なアドレスのみがアクセスされる。各バストランザクションの数(転送されるワードの数即ち「バースト寸法」)は異なる場合があり且つフレームデータ転送及び統計ダンプに対しプログラム可能である。管理ブロック200及び記述子転送寸法は必要に応じてネットワーク制御器40によって決定され且つ1乃至32の2個の連続したワードの範囲とすることが可能である。システムバスインターフェースユニット80内部のDMAユニット85は制御器によるシステムバスの利用を最小とするために単一サイクルアクセスに対する必要なタイミングを与える。

【0053】ネットワーク制御器40に対するコンフィギュレーション/ステータスレジスタアクセスは、DMA転送に対して使用されるものと同一の32ビットデータバスを使用して行うことが可能である。そのために、制御器がバスマスタである場合にはレジスタアクセスを実施することは不可能である。コンフィギュレーション/ステータス(省略して「config」と言う)動作は殆どのポピュラーなマイクロプロセッサと共に動作するように設計されている。ネットワーク制御器内部の全ての位置は32ビットレジスタとして実現することが可能である。全てのコンフィギュレーション(形態特定)及びステータスレジスタは、ネットワーク統計の全てと共に、このインターフェースを介してアクセスすることが可能である。

【0054】次に、図4を参照すると、本発明の制御器の動作は3つの重要なシステムメモリデータ構造、即ち(1)管理ブロック200、(2)記述子206を有する記述子リング202、(3)フレームデータバッファ204が関与する。任意の与えられた適用例に対して、

10

20

30

40

50

1 個の管理ブロック 2 0 0、8 個の記述子リング 2 0 2 (図 3)、複数のフレームデータバッファ 2 0 4 が使用される。図 3 に示したように、各ポートにおいて各 F I F O 7 0、7 2 に対して 1 個の記述子リング 2 0 2 が存在している。制御器 4 0 を初期化する前に、ホスト 4 4 は、システムメモリにおいてこれらのデータ構造を割当て且つ形態特定することが期待される。管理ブロック 2 0 0 は、チップ初期化のため及び制御器によって維持されているネットワーク統計に対する交換点として使用される。

【0 0 5 5】各記述子リング 2 0 2 は、当業者にとって公知の如く、フレームデータバッファ 2 0 4 に対する情報及びポインタを有するエントリ又は記述子 2 0 6 を具備する循環型のキュー即ち待ち行列である。これらの記述子及び記述子リングを使用することを示した装置及びシステムの例は米国特許第 5、2 9 9、3 1 3 号及び第 5、1 3 6、5 8 2 号に開示されており、それらの開示を引用によって本明細書に取込む。各記述子リング 2 0 2 は制御器 4 0 内の特定の F I F O 7 0、7 2 に対して専用とされており、且つ 1 個のリング内の各 2 ワード記述子エントリ 2 0 6 はシステムメモリ内の 1 つの特定のフレームデータバッファ 2 0 4 と関連している (図 5)。データバッファは送信用のフレームを包含するか又はフレーム自身のためのスペースを与えるメモリのブロック (典型的に、5 1 2 乃至 2、0 4 8 バイトの範囲) として定義される。

【0 0 5 6】制御器 4 0 の初期化の一部として、ホストはシステムメモリの 1 つのセクションを取り分けねばならない。このメモリはバッファ管理ポインタ、コンフィギュレーション情報及びポート当たりのネットワーク統計を保持するために使用される。管理ブロック 2 0 0 は統計で周期的にアップデートすることが可能であり且つ制御器 4 2 によって参照されるので、それは装置の動作全体にわたりメモリのアクティブ即ち活性状態にある割当てに留まらねばならない。

【0 0 5 7】管理ブロック 2 0 0 (初期化ブロックとも呼ばれる) は 5 1 2 個の隣接したバイトから構成されており且つメモリ内においてワード整合されている。図 7 は管理ブロック 2 0 0 及びその詳細をより詳細に示している。管理ブロックの最初の 1 5 個のワード 2 0 0 a はチップ初期化のために使用される情報を有している。該制御器は、常に、共用システムメモリ 4 6 から再度一部又は全てをフェッチすることの命令が与えられない限り、このセクションのオンチップコピーを参照する。管理ブロック 2 0 0 の初期化セクション 2 0 0 a は 8 個の記述子リング 2 0 2 に対するシステムメモリポインタ、及び 6 個のオンチップタイマに対するセットアップ情報及び 9 個の DMA バスマスタバースト寸法 (バス所有当たりに種々のタイプのデータに対し転送されるワードの最大数) を包含している。次の隣接する 4 個のワード 2

0 0 b は、以下に説明するように、記述子リング 2 0 2 の幾何学的形状及び外部共用メモリ 4 6 における関連するフレームデータバッファ寸法を画定するためにホスト 4 3 によって使用することが可能である。制御器 4 0 は自動的に (送信) T X 及び (受信) R X 記述子リング 2 0 2 を構築することが可能である (図 3)。

【0 0 5 8】管理ブロック 2 0 0 の残りのワード 2 0 0 c は適宜のプリミティブによってそのように命令が与えられる場合に、そのオンチップ H D L C フレーム統計のイメージを共用システムメモリ 4 6 内にコピーするために制御器 4 0 に対しスペースを与える。これらの周期的な統計のスナップショットはシステムが使用するためである。管理ブロック 2 0 0 のこれらのワードの割当ては、統計ダンプ特徴が使用されない場合には要求されることはない。

【0 0 5 9】チップリセットが完了した後に、リセット進行中ピンが不活性状態となると、図 4 5 及び 4 6 に示してあり且つセクション V を参照して以下に詳細に説明するように、初期化手順が開始することが可能である。最初に、ホストが管理ブロック 2 0 0、記述子リング 2 0 2、システムメモリ内のフレームデータバッファ 2 0 4 をセットアップする。2 番目に、ホスト 4 4 は管理ブロック 2 0 0 の開始システムアドレスを「管理ブロックに対するポインタ」 (P A B) と呼ばれる制御器 4 0 内部のレジスタに対して管理ブロック 2 0 0 の開始システムアドレスを書込み、且つ、オプションとして、プリミティブインタラプトをイネーブルさせる。次いで、インタラプト (I N T) プリミティブがホスト 4 4 によってネットワーク制御器に対して発行される。このことは、制御器をして、管理ブロック 2 0 0 の最初の 3 2 個のワード (図 7) を処理を行うためにネットワーク制御器のチップ内にコピーさせる。次いで、ネットワーク制御器はアクノレジメント I N I T _ C O M P L E T E 又は A C T (I N I T) プリミティブインタラプトでホストに対して応答する。この点において、ホスト 4 4 はハウスキーピングを行うか又は制御器のレジスタの全てのコンフィギュレーション即ち形態特定を行うことが可能であり、各 H D L C ポートに対する動作モードを確立し、送信器及び受信器をイネーブルし、且つ種々のインタラプトをイネーブル及びマスクすることが可能である。図 4 5 に更に詳細に示されているように、完了すると、ホストはネットワーク制御器 4 0 に対して S T A R T (開始) プリミティブを発行し、通常の動作を開始させる。S T A R T プリミティブは制御器をして、8 個の送信及び受信記述子リングのうちの各々における最初の 2 つの記述子をプリフェッチさせ且つフレーム転送の準備を行わせる。

【0 0 6 0】管理ブロック 2 0 0 内の最初の 8 個のエントリは各記述子リング 2 0 2 (図 3) のトップ即ち一番上のものに対するポインタとして作用するシステムアド

レスである。記述子 2 0 6 はメモリ内においてワード整合（即ちバイト整合）されていなければならないので、これらのポインタは、常に、最小桁の 2 つのアドレスビット（バイトアドレス）においてゼロでプログラムされるべきである。換言すると、全ての記述子リングポインタは 4 によって均等に割算可能なものとすべきである。整合されていない記述子リングポインタアドレスから予測不可能な動作が発生する。ネットワーク制御器 4 0 は I N I T プリミティブが完了するとこれらのポインタの 10 コピーを参照し、別の I N I T が実施されるか又はリフレッシュ記述子リングプリミティブが発行されない限り、I N I T が影響を有することのない後にシステムメモリ内のポインタを変化させる。

【0 0 6 1】前述したように、各ポート 5 0, 5 2, 5 4, 5 6 内の各送信チャンネル及び各受信チャンネルは全部で 8 個のリングに対する専用の記述子リング 2 0 2 を使用する（ポート当たり 1 個の送信リング及び 1 個の受信リング）（図 3 及び 4）。記述子リング 2 0 2（図 4）は「記述子 2 0 6」と呼ばれる幾つかの 2 ワードエントリから構成される循環型のキュー即ち待ち行列である。各記述子エントリ 2 0 6 は 1 個のフレームデータバッファ 2 0 4 を記述する。記述子 2 0 6 エントリの最初のワード 2 0 8 はそのフレームデータバッファ 2 0 4 に関する情報及びそのフレームデータバッファが収容しているフレーム又は部分的なフレームを包含している（図 5）。記述子 2 0 6 エントリの 2 番目のワード 2 1 0 はシステムアドレスであり、即ちその関連するフレームデータバッファのトップ即ち一番上に対するポインタである。記述子リング 2 0 2 は 1 乃至 8 K 個のエントリの寸法の範囲とすることが可能である。ネットワーク制御器 4 0 は初期化において管理ブロック 2 0 0 における各 30 リングのトップ即ち一番上に対するポインタが与えられる。記述子エントリ 2 0 6 は、常に、リングのトップ即ち一番上から初めて逐次的にアクセスされる。記述子リング 2 0 2 内の最後の記述子はリングの終りであることを示すフラグを有している。該制御器は、それがリングの終りフラグに遭遇すると、そのリング内の最初のエントリへリターン即ちラップする。

【0 0 6 2】各記述子 2 0 6 の最初のワード内の所有ビット（O B）2 1 2 は、ホスト又は制御器が関連するフレームデータバッファを所有しているか否かを表わす。所有は、制御器及びホストによって準拠されねばならない特定のプロトコルに従う。その規則は簡単である。記述子 2 0 6 の所有が他のものに対して放棄されると、その記述子又はそれと関連するバッファのどの部分も変更されることはない。ホストはフレーム受信のための空のバッファ及びフレーム送信のための満杯のフレームデータバッファの所有を制御器に与える。逆に、ネットワーク制御器はそれが使用した送信バッファ及びそれが満杯とさせた受信バッファについて所有をホストへ戻す。 50

【0 0 6 3】任意の与えられたポート上でのフレーム受信の場合、ホスト 4 4 は空のフレームデータバッファ 2 0 4 に対してポイントする連続した記述子の所有を制御器 4 0 へ供給することが要求される。そのフレームの正に最初のワードがメモリ 4 6 へ転送された後に、フレームアドレス通知（F A N）インタラプトが発行される

（図 1 3 - 2 1 を参照してセクション I において後に更に詳細に説明する）。制御器によって 1 個のフレームが完全に受信されると、その構成要素である記述子の所有は再度割当てられる。ホストは、R I N T インタラプトを介してこのイベントに関しての信号が与えられる。ホスト 4 4 は、その信号を発行する特定のポートを推測するためにマスタインタラプトレジスタ（M I R）（図 8 B）を読むことを余儀なくされる。このことが達成されると、そのフレームは何等かの態様でディスパッチ即ち送給することが可能であり且つ関連する記述子の所有は制御器へリターンされる。

【0 0 6 4】典型的な動作においては、ホスト 4 4 は記述子リング 2 0 2 の周りをネットワーク制御器 4 0 に「追従」し、ネットワーク制御器 4 0 が使用するのためにその後に「空の」バッファ記述子 2 0 6 を残存させる。ネットワーク制御器 4 0 がホスト 4 4 の前方に離れ過ぎると、それは記述子リング 2 0 2 をラップアラウンドし且つそれが所有することのない記述子 2 0 6 に遭遇する場合がある。このことが発生すると、入ってくるフレームが失われる場合がある。ホストは、早期輻輳通知（E C N）インタラプトを介して受信 F I F O 7 0 のオーバーフローが通知される（図 2 6 - 4 3 を参照してセクション I I I において後に更に詳細に説明する）。次いで、ホストは、更なるフレームの喪失を回避するためにその動作を変更すべく対応することが可能である。

【0 0 6 5】与えられたポート上でのフレーム送信の場合、ネットワーク制御器 4 0 は送信記述子リング 2 0 2 の周りをホスト 4 4 に「追従」し、ホストが獲得するためにその後に使用済みのバッファ記述子を残存させる。ホストは 1 個又はそれ以上のフレームが送信の準備がなされている場合に制御器 4 0 に対して記述子 2 0 6 の所有を与えるに過ぎない。1 つのフレームが制御器によって完全に送信されると、その構成要素の記述子 2 0 6 の所有は再度使用するためにホスト 4 4 へ戻される。ホスト 4 4 は、このイベントについて T I N T インタラプトを介して信号が与えられる。

【0 0 6 6】幾つかの適用例においては、ホスト 4 4 は受信されるか又は送信されたフレームよりも寸法がより小さなフレームデータバッファ 2 0 6 を使用することを選択する場合がある。単一のフレームは複数個のバッファをスパン、即ち複数個のバッファにわたる。このことは、ネットワーク制御器 4 0 によってフレームを切断（受信時に散乱させる）又は組立（送信時に収集する）50 することを可能とする。複数個のデータバッファは、関

連する記述子 2 0 6 を一緒に「チェーン化」することによって 1 個のフレームの構成要素の断片を保持することが可能である。定義上、チェーン化した記述子はそのチェーンの終端の記述子においてフレームの終り (E O F) フラグ 2 1 4 がセットされている記述子リング内の連続したエントリである。換言すると、所有されているがフレームの終りフラグがセットされていない記述子エントリのバッファは 1 つのフレームの一部であって、全体的なフレームではないと考えられる。

【0 0 6 7】大きなフレームの受信期間中に、ネットワーク制御器 4 0 はそれが各フレームデータバッファ 2 0 4 を完全に充填すると、1 個ずつ、記述子 2 0 6 を一緒にチェーン化する。フレームの終りが受取られ且つシステムメモリへ転送されると、フレームの終りフラグ (E O F) がそのチェーンの終端記述子内にセットされる。送信期間中に、ネットワーク制御器 4 0 はチェーン化されたバッファの内容から単一のフレームを逐次的に構築することが可能である。フレームの送信は、その記述子がセットされているフレームの終りフラグを有しているバッファに遭遇する場合にのみ終了する。

【0 0 6 8】ネットワーク制御器 4 0 は、関与する最初の及び最後の記述子エントリをアップデートすることによって 3 個又はそれ以上のフレームデータバッファと一緒にチェーン化される場合にバスの利用を最適化させる (図 4)。ネットワーク制御器 4 0 がチェーン化されているフレームに関与するバッファについて終了すると、それは、最初に、最後の記述子の所有をリターンし、次いで、それは最初の記述子の所有をリターンする。これらはフレームの「フェンスポスト (fencepost)」即ち垣根の支柱である (図 4 4 及びセクション I V において以下に説明する)。ホスト 4 4 は、例えばそれらが制御器によって所有されているものであっても、全ての中間のフレームデータバッファの所有を獲得する。従って、ホストがフレームの終りフラグによってマークが付けられていないホストが所有している記述子に遭遇すると、それは、フレームの終りフラグがセットされている次のホストが所有している記述子に至るまで且つそれを包含する全ての次続の記述子の所有を獲得する。

【0 0 6 9】「フェンスポスト」即ち垣根の支柱型チェーン内の最初及び最後の記述子のフラグ及びフィールドの全ては制御器 4 0 によってアップデートされて、それが完全に送信されるか又は受信されるとフレームに関しての正確な情報を提供する。記述子 2 0 6 の最初のワード 2 0 8 は、更に、バッファ寸法 2 1 6 及びメッセージ寸法 2 1 8 を有している。受信フレームの場合には、チェーン内の最初の記述子のメッセージ寸法 2 1 8 (M S I Z E) フィールドは、単に、関連するフレームデータバッファのバイトカウント (何故ならば、これはバッファ寸法に等しいからである) ではなく、全体的なフレームのバイトカウントでアップデートされる。然しながら

ら、終端記述子のメッセージ寸法フィールド 2 1 8 はその関連するバッファ内のフレームデータによって占有される実際のバイト数のみを有している。このことは、バッファ内のフレームデータに続く最初の完全なワード内に格納されている受信ステータスワードをホストが容易に捜し出すことを可能とする (注意すべきことであるが、ステータスワードの 4 バイトは M S I Z E フィールド内に格納されているカウント内に含まれてはいない)。

【0 0 7 0】単一のフレームデータバッファ 2 0 4 内には 1 つを超えるフレームが存在すべきではない。単一のフレームは、記述子が記述子リング 2 0 2 において隣接したものの即ち連続的なものである場合には、複数の記述子 2 0 6 のフレームデータバッファ 2 0 4 をスパンする、即ちそれらにわたることが可能である。このことはバッファチェーン化と呼ばれる。ネットワーク制御器 4 0 は、常に、幾つかの空で且つ隣接した即ち連続した受信バッファの所有を有するべきである。ネットワーク制御器 4 0 は、送信の準備がなされているフレームを包含する送信バッファの所有のみが与えられるべきである。

【0 0 7 1】必ず必要とされるわけではないが、フレームデータバッファ 2 0 4 がメモリ内においてワード整合されており且つチェーン化が必要とされることがない程大きいものである場合に最良の性能が得られる。

【0 0 7 2】典型的な「ストアアンドフォワード (store-and-forward)」即ち「格納及び転送」適用例においては、ホストはシステムメモリ内の空で割当てられていないフレームデータバッファ 2 0 4 の「プール」を維持する。受信記述子 2 0 6 に対してフレームデータバッファ 2 0 4 を割当ててことは、実効的に、それをこのプールから除去する。フレームデータバッファ 2 0 4 が充填されると即ち満杯になると、それは 1 つ又はそれ以上の送信記述子へ再割当て即ちスイッチされる。送信が完了すると、フレームデータバッファ 2 0 4 は再使用のためにプールへリターン即ち戻され且つサイクルが繰返される。

【0 0 7 3】記述子リングポインタ 2 0 0 d の後の管理ブロック 2 0 0 内の次の 2 つのワードはタイマ再ロード及び制御情報 2 0 0 e を包含している。該制御器は、U C L K 周波数 2 2 4 を分割するために分割器 2 2 2 及びハードウェアプレスケールタイマ 2 2 0 (図 6) を使用する。プレスケールタイマ再ロード値 2 2 6 は、プレスケールタイマの出力周波数を調節するために使用される。典型的に、プレスケール再ロード値は、2 0 ミリ秒 (5 0 H z) プレスケールタイマ周期となるように選択されるが、より速い及び遅い周期も可能である。プレスケールタイマ 2 2 6 の出力はネットワーク制御器 4 0 内部に維持されている幾つかの二次的 8 ビットタイマ 2 2 8 に対するベースのインクリメント周波数として使用される。これらの二次的タイマは、統計ダンプタイマ 2 3

25

0、ポート0-3送信記述子プールタイマ(4)232-238とすることが可能である。5個の8ビットタイマの各々は管理ブロック200内において確立される関連する再ロード値を有している。以下の式はどのようにしてプレスケールタイマ再ロード値を計算するかを示している。

*

表1:典型的なプレスケールタイマ再ロード値

f_{CLK} (MHz)	T_{CLK} (ns)	10進数再ロード値 (20ms)	16ビットHex再ロード値 (20ms)
33	30	23.869	0x5D3D
25	40	34.286	0x7EE6

次の式は二次的タイマ再ロード値をどのようにして計算するかを示している。

【0076】二次的再ロード = $265 - (T_{secondary} / T_{prescale})$

*

表2:典型的な二次的タイマ再ロード値

$T_{prescale}$ (ms)	$T_{secondary}$ (秒)	10進数再ロード値	8ビットHex再ロード値
20	0.5	231	0xE7
20	1.0	206	0xCE
20	2.0	156	0x9C
20	5.0	6	0x06

二次的タイマの各々は管理ブロックのフィールドをイネーブルするタイマ内に含まれている対応するイネーブル制御ビットを有している(図7)。「1」がタイマをイネーブル即ち動作可能とさせ、「0」がタイマをディスエーブル即ち動作不能状態とさせる。以下の表は5個の二次的タイマイネーブルの各々のビット位置を示している。該制御器は、INITが完了すると、これらのイネーブルのオンチップコピーを参照する。システムメモリ

★内のイネーブルを変化させることは、別のINITが実施されるか又はTIMER_ENABLEプリミティブが発行(0x0F)されない限り、何等影響を有するものではない。プレスケールタイマは、二次的タイマのいずれもがイネーブルされない場合には自動的にディスエーブルされる。

【0078】

表3:管理ブロックタイマイネーブルフィールド

(1=イネーブル;0=ディスエーブル)

ビット	7	6	5	4	3	2	1	0
名称	予約	予約	予約	ダンプ開始Tx		Tx	Tx	Tx
						3ボール	2ボール	1ボール0ボール

プレスケールタイマ220の構造は、広範囲のタイマ分解能を可能とする。プレスケールタイマ再ロード値226を選択する場合に、各プレスケールタイマの期間満了は制御器のオンチップ処理帯域幅の小さな割合部分を消費する。非常に小さなプレスケールタイマ周期(大きな再ロード値)を選択することは、入って来るフレーム及び出て行くフレームをサービスするための制御器の能力に対して不本意に悪影響を与える場合があり、それにより装置の全体的な性能に影響を与える場合がある。プレスケールタイマは1ミリ秒周期未満で動作しないようにすることが推奨される(図6)。

【0079】送信記述子ボールタイマ232-238に対する二次的タイマ再ロード値を選択する場合に、2つのファクタが考慮されるべきであり、即ち、(1)ポートの半二重又は全二重動作モード、及び(2)与えられ

50

26

*【0074】プレスケール再ロード = $65.536 - (T_{prescale} / (16 \times T_{CLK}))$

尚、 $T_{prescale}$ は所望のプレスケールタイマ周期であり且つ T_{CLK} はシステムクロック周期である。

【0075】

※尚、 $T_{secondary}$ は所望の二次的タイマ周期であり且つ $T_{prescale}$ はプレスケールタイマ周期である。

【0077】

選択されたバースト寸法より小さなものでない限り、これらのフィールド内に設定されている値によって決定されるバースト寸法でデータを転送する。該制御器は、INITプリミティブが完了すると、これらの値のオンチップコピーを参照する。その後の変化は適宜のプリミティブコマンド（基本命令）の提出を介して表わされねばならない。

【0081】バースト及びフレームバッファ寸法を等しく設定することは、フレーム当たりの必要とされるバス転送の数を最小とさせ且つシステムの拘束条件が大きなDMAバーストを許容する場合には、改良した性能を提供する。

【0082】システムクロック周期200gはPAB+48のバイト#1内に位置されており、25MHzで動作する場合には「0x28」の値を有し、また33MHzのシステムクロックで動作する場合には「0x1E」の値を有するべきである。該制御器は、INITプリミティブが完了されると、この値のオンチップコピーを専ら参照し、別のINITが実施されない限り、INITが何等影響を有することがない後にシステムメモリ内におけるこの値を変化させる。

【0083】「N1」は受信されるべき最大フレーム寸法に対しホストによって選択可能な16ビット変数である。ポート#0及び#1に対するN1の値はPAB+52200hに位置されており且つポート#2及び#3に対する値はPAB+56200iに位置されている。このN1の値は、典型的に、初期化時にホストによってプログラムされ且つ1バイト乃至64Kバイトの間の範囲内とすることが可能である。典型的に、N1は殆どの適用例に対し2Kバイト又はそれ以下である。N1を超える受信フレームはそのポートに対する「Frames Larger Than N1 (N1より大きなフレーム)」統計をインクリメントさせる。制御器40は、INITプリミティブが完了すると、これらの値のオンチップコピーを参照し、別のINITが実施されない限り、INITが影響を有することのない後のシステムメモリにおけるこれらの値を変化させる。

【0084】ネットワーク制御器40は、「Transmit (TX) Ring Size (送信 (TX) リング寸法)」又は「Receive (RX) Ring Size (受信 (RX) リング寸法)」フィールド(PAB+60乃至PAB+72)200bの値がゼロでない場合には、共用メモリ46内に特定の送信及び／又は受信記述子リング202を自動的に構築する。そうでない場合には、これらのフィールドはゼロであり、制御器ファームウェア102は関連する記述子リングを構築することではなく、その代わりに、ホスト44が共用メモリ46内にこれらの構造を既に構築したものと期待する。

【0085】プリミティブコマンドレジスタ(PCR)

(図8A)は処理のためにネットワーク制御器40の内部ファームウェア102に対してホストのシステムソフトウェアがコマンド／命令を発行するためのメカニズムを提供している。発行される各々及び全てのホストプリミティブ(このレジスタの下半分におけるもの)はプロバイダプリミティブ(このレジスタの上半分におけるもの)を介してファームウェアによってアクノレージメント即ち受取りの確認が行われる。

【0086】プリミティブ交換プロトコルは、プリミティブメカニズムが適切に動作するためにホスト及びファームウェアの両方によって従われねばならない。ホストは一度に1個のみのプリミティブを発行するものでなければならず、別のプリミティブを発行する前にプロバイダプリミティブのアクノレージメント即ち受取確認を待機せねばならない。他方、ファームウェアは発行された各ホストプリミティブに対してただ1つのプロバイダプリミティブを発生する。

【0087】マスタインタラプトレジスタ(MIR)

(図8B)はMINTR#ピンを介してホストプロセサに対して報告するためのイベントを記録する。該レジスタはバイトの位置に関する一貫性のために分配される幾つかの雑多なビット(即ち、PINT, SPURINT, MERR, PPLOST, SERR, HPLOST, WERR)と共に、HDL Cポート当たりほぼ1バイトのインタラプトイベントに組織化される。

【0088】例えばマスタインタラプトマスケレジスタ(MIMR)及びポートエラーインタラプトマスケレジスタ(PEIMR)等のその他の詳細には説明しないレジスタは、ホストが、どの対応するMIR及びPEIRインタラプトイベントが実際に種々のピン上でインタラプトを発生するかを選択することを可能とする。これらのレジスタはMIR及びPEIR内のビットの設定に影響を与えるものではなく、それらは、単に、インタラプトビットが送られることの結果としてホストインタラプトの発生をマスクするのに過ぎない。

【0089】I. フレームアドレス通知(FAN)

図9-21に戻ると、クラシックな格納及び転送(SF)アーキテクチャとカットスルー(C/T)アーキテクチャとの間のハイブリッドオプションを可能とするフレームアドレス通知(FAN)インタラプトを示した詳細図が示されている。本発明によれば、フレームアドレス通知(FAN)は、受信したフレームに対する全ての関連するアドレスフィールドが現在共用メモリ46内に存在している場合に、ホストプロセサ44へ信号が送られるインタラプトである。次いで、該フレームは適宜のアルゴリズム及びルックアップテーブル46c(図20)と共にアドレス及びルックアップエンジンによって処理し且つ適切なポート及びデスティネーションヘイスパッチ即ち送給することが可能である。このことはパイプライン効果を与える。何故ならば、ルーチング

は、フレームの残部がネットワークワイヤから入って来ることを可能としながら並列的に行うことが可能だからである。

【0090】更に、DMA 85 のバースト寸法を注意深く選択することによって、最初のバーストがフレームから読取られる場合に適宜のアドレスフィールドを使用可能とさせることが可能である。MAC レベルヘッダ、IP アドレス、又は例えば TCP/UDP ポートであっても、バーストの寸法に依存してメモリ内に読込むことが可能である。このことは L 2 - L 3 又は L 4 フレームス
10 イッチング適用例を容易なものとさせる。

【0091】図 9、10、11、12 はどのようにして TCP/UDP ヘッダが IP データエリア内にカプセル化され且つ IP ヘッダが MAC データエリア内に包含されているかを示している。図 9 はレイヤ構成の良好な表示を与えている。TCP/UDP データエリア 240 及び TCP/UDP ヘッダ 240a、IP データエリア 242、ヘッダ 242a、MAC データエリア 244 及び MAC ヘッダ 244a が示されている。

【0092】図 10 は 18 バイトの 802.3 (MAC) データリンクレイヤヘッダを示しており、一方 20 バイトインターネット IP ヘッダは図 11 に示されている。図 12 は 20 バイト TP ヘッダを示している。適宜のアドレスフィールドがリストされている。

【0093】図 13 - 20 は本発明に従ってデータフレームのルーチング即ち経路付けを行う方法及びシステムの基本的なプロセスを示している。図示したように、SWIFT の符号を付けたネットワーク制御器 40 は 4 個の HDLC ポート 50、52、54、56 を有しており、各ポートは送信 FIFO 70 及び受信 FIFO 72
30 を有している。ネットワーク制御器は、更に、制御プロセサ (CPC) 92 としても知られている RISC プロセサ及び直接メモリアクセスユニット (DMA) 85 を有している。CPC バス 250 は CPC 92 と DMA 85 ユニットとの間を相互接続する。インタラプトバス 252 は種々の HDLC ポートと CPC 92 との間を接続する。FIFO バス 254 は DMA と種々の HDLC ポートとの間を相互接続する。

【0094】図 14 に示したように、フレームは最初に HDLC ポート 3 へ入り且つネットワーク制御器 40 の
40 受信 FIFO 72 内に受信される。図 14 において、そのフレームは矢印 258 で示されたウォーターマークに到達し、且つそのポートはインタラプトバス 252 を介して CPC 92 に対しパケットの開始 (SOP) インタラプト (図 15) を開始させる。この時に、該フレームからのデータがいまだに FIFO 72 内へ転送されている間に、CPC 92 はデータを転送するために DMA 85 (図 16) に対するコマンドを発行する。DMA 85 はシステムバス 42 を介してバスアービトリション
(調停) 論理ユニット 47 に対してクエリ (質問) を
50

発行し、それがシステムバス (図 17) を使用することが可能であるか否かを質問する。システムバス 42 が使用可能である場合には、バスアービトリション論理ユニット 47 は肯定状態に入る。同時に、フレームは FIFO 72 内に継続して受信される。この時に、DMA 85 は FIFO 72 からのデータを図 18 に示したように共用システムメモリ 46 へ転送する。図 18 に示したように、この DMA 85 の最初のバーストは、次いで、CPC 92 をして、FAN 又はフレームアドレス通知イベントとして知られているインタラプト信号をシステムバス 42 を介してホストプロセサ 44 に対して発行させ、フレームの予め選択されたアドレスフィールドが共用メモリ 46 (図 19) 内に存在することを表わす。DMA
バースト寸法の量は、見られる特定のヘッダ及びアドレスに対して且つどのレイヤに対するものであるかに対して調節されている。

【0095】図 20 に示したように、次いで、ホストプロセサ 44 はルックアップアルゴリズムを開始し且つどのようにしてパケット及びフレームがアドレスされ且つ転送されるかを決定する。このルックアップ及び FAN イベントは、フレーム受信バッファ内にフレームが未だ受信中である場合であっても発生することが可能である。

【0096】共用メモリ 46 内に 1 つのフレームが完全に受信されると、フレームの終り (EOF) インタラプトが発行される。従って、このことは、ホストが転送又は転送プロセスを終了することが可能であることを意味する。

【0097】図 21 はフレームアドレス通知 (FAN) イベントを示したタイミングチャートを示している。MAC レイヤと共に一番上に示したように、P1 として示されているパケットの始めが最初に発行され、次いで DMA に対するファームウェア (FW) 命令が発行されて受信器とのパケットの開始コマンドを構築する。パケットの継続 (COP) コマンドが発行され、次いで、図示したように、DMA がデータを転送する。DMA は、又、フレームアドレス通知を発行し、次いで、パケットの終り (EOP) を発行する。MAC レイヤの一番う
えに示したように、P2 として知られる第二パケットの場合にも同様の状態が発生する。

【0098】II. ルックアヘッドウォーターマーク
次に、図 22 - 25 を参照すると、本発明において使用されるルックアヘッドウォーターマーク (look-ahead watermark) のより詳細が示されている。ルックアヘッドウォーターマーク (LAWM) は同期信号として機能し、その場合に送信及び受信 FIFO 70、72 を有する FIFO (先入先出) メモリが 1 個又はそれ以上の付加的な書込バーストを受信するために充分な格納領域が存在していることを表わすためにルックアヘッドウォーターマーク (LAWM) を供給す

る。フレームの送信は、この技術によって促進させることが可能である。何故ならば、それは通信プロセッサ 9 2 上の負荷を減少させる一方バス及びメモリ資源の利用を増加させるからである。

【0 0 9 9】ルックアヘッドウォーターマーク信号は、F I F O が表示された量の付加的な DMA バーストを受け付けることが可能であることを暗示する。DMA バースト寸法はルックアヘッドウォーターマークが取り次いだバーストと同一の寸法であることが必要とされるものではない。ルックアヘッドウォーターマークは、従来の送信「レベル感応性」ウォーターマークメカニズムとしてよりも「容量インジケータ」として機能する。別の観点においては、ルックアヘッドウォーターマークはスタンダードの「ボトムアップ」ウォーターマークではなく「トップダウン」容量インジケータである。

【0 1 0 0】ルックアヘッドウォーターマークは利点を有しており且つデータの処理を助ける。それは、F I F O アンダーフローエラーを減少させるか又は取除くことを可能とする。それは直接メモリアクセスユニットの利用を改善する。それは、又、フレーム転送を促進させる。それは送信用の次のフレームの早期の検知を可能とする。それは高価な F I F O メモリ利用を改善し且つネットワークのフレーム間ギャップタイミング「遅延」を減少させる。それは、又、フレーム当たりのサイクル、即ちマイクロプロセッサの作業負荷を減少させることを可能とし、且つ小型及び大型の両方のフレームに対し効率を向上させることを可能とする。それはホストシステムに対してトランスペアレント即ち透明であり且つ C P U コンテキストスイッチング即ち文脈上のスイッチングを減少させる。

【0 1 0 1】ルックアヘッドウォーターマークは該装置（ファームウェア／ハードウェアステートマシン）がそれが付加的なデータのバースト（既知の量の）をサポートし、従ってフレーム当たり 1 個又はそれ以上の C P U コンテキストスイッチを除去、減少することが可能であるかを決定するために F I F O メモリ内を「見る」ことを可能とする。2 番目の DMA コマンドは、次のフレームバーストをデスティネーション F I F O へ移動させるために殆ど付加的なオーバーヘッドなしにキュー化即ち待ち行列化させることが可能である。

【0 1 0 2】図 2 2 はルックアヘッドウォーターマークと従来の F I F O フロー制御とを示している。この図はシステム側とネットワーク側を示した F I F O メモリ構造の基本的概念の抽象的な描写である。送信ウォーターマークは 2 6 0 において示されている。タイミングメカニズムは底部の水平線上に示されており且つデータバースト X に対して点 1 において示されたデータバースト及び点 2 及び 3 におけるルックアヘッドウォーターマークデータバースト Y を有する時間を示している。ルックアヘッドウォーターマーク時間線はファームウェアルック

アヘッドウォーターマークチェックを示している。従来例においては、F I F O は空（データ＝0）であり、次いで、インタラプトが発生され且つ次いで現在のデータが X であるように 1 つのデータバーストが F I F O を充填する。ファームウェアルックアヘッドウォーターマークチェックの場合には、ファームウェアは F I F O へのデータ転送のために DMA に対してコマンドを提出し且つ 2 番目のデータバーストは数字 2 によって示されるように発生し且つ現在のデータは $X + Y$ となる。次いで、ファームウェアはルックアヘッドウォーターマークをチェックし且つ 3 番目のデータバーストは数字 3 によって示されるように発生し、現在のデータは $X + 2 Y$ となる。

【0 1 0 3】図 2 3 におけるフローチャートに示したように、ブロック 3 0 0 において開始するルックアヘッドウォーターマークを有する本発明のネットワーク制御器を使用してデータをベースとしたネットワークにおけるデータの流れを制御する本発明方法が示されている。ブロック 3 0 0 において、DMA バースト寸法が格納され、且つルックアヘッドウォーターマークバースト寸法も格納される。2 つのバースト寸法は実質的に同一のものとするか又は異なるものとするが可能である。次いで、該チャンネルがイネーブル即ち動作可能状態とされる。次いで、ウォーターマークインタラプトがブロック 3 0 2 において DMA に対して発生される。ブロック 3 0 4 において、ファームウェアは DMA に対するデータ転送コマンドを発行する。このコマンドの一部として、ファームウェアはそのタスクが完了すると、即ち R E O C = T R U E であると、コマンドの終りに対する要求（R E O C）を介してアクノレジメント即ち受信確認を DMA に対して要求する。ブロック 3 0 6 において、DMA は拡張バスに対する調停を行い、次いで、送信 F I F O へデータを転送する。それは E O C フラグを介して、いつそれが終了するかを知らせる。

【0 1 0 4】DMA 転送が完了したか否か、即ち E O C = T R U E に対応するか否かを決定するための判別がブロック 3 0 8 において行われる。DMA 転送が完了していない場合には、ブロック 3 0 6 が繰返される。DMA 転送が完了している場合には、F I F O 制御論理はブロック 3 1 0 においてデータ容量を決定する。図示した如く、F I F O 制御論理は、F I F O 内に保持されている現在のデータ値を F I F O 内に保持することの可能な最大値から減算することによってデータ容量を計算する。その結果は、ルックアヘッドウォーターマークバースト寸法によって割算されデータ容量が得られる。ブロック 3 1 2 に示したように、データ容量が 1 より大きい或又はそれと等しい場合には、ルックアヘッドウォーターマーク値（例えばフラグ）は真（T R U E）である。ルックアヘッドウォーターマーク値が 1 未満である場合には、それは偽（F A L S E）である。ルックアヘッドウ

ウォーターマークフラグがブロック314において真であると、付加的なコマンドがブロック316においてDMAに対して発行され、且つDMAはブロック318において送信FIFOヘデータを転送する。ルックアヘッドウォーターマークが偽であると、そのルーチンは終了する。

【0105】図24a及び24bは最初にインタラプト取次フレーム送信(図24a)及びルックアヘッドウォーターマーク取次フレーム送信(図24b)を示している。これらのタイミングメカニズムはルックアヘッドウォーターマークの利点を示しており且つクロックサイクルによってルックアヘッドウォーターマークの効率を量子化する場合の助けとなる。これらのチャートはインタラプトのずらされた遅延を示しており、例えばそれらがいつ発行され且つサービスされるか且ついつデータがFIFO内に書込まれるかを示している。このことはそれが完全に使用されることを確保するためにビジーなマルチチャンネル装置において重要である。このことはスタンダードのインタラプトの待ち時間をルックアヘッドウォーターマークの効率と比較することを可能とする。

【0106】インタラプト取次フレーム送信(図24a)

1. DMAがパケットインタラプト信号の開始(SOP)を介してフレーム送信を開始する。

【0107】2. ファームウェア(FW)が送信チャンネルをイネーブルさせ、コマンド(2個の32ビットワード)を構築し且つこのコマンドを実行するためにDMAへ提出する。

【0108】3. DMAが該コマンドをデコードし、外部バスに対する調停を行い、外部共用メモリから適宜のデータを読み取り、これを適宜の送信FIFOメモリ内に書込む。

【0109】4. DMA転送が完了した後で且つ送信ウォーターマークが超えられていない場合に、パケットの継続(COP)インタラプトが発生される。

【0110】5. 再度、ファームウェアはコマンドを構築し且つそれを実行のためにDMAへ発行する。

【0111】6. ファームウェアが該COPインタラプトをディスエーブルしておらず且つ該FIFO内のデータがスタンダードのウォーターマークを超えていない場合には、別のCOPを発生することが可能である。

【0112】7. 該フレームの終端バイトがFIFOからネットワーク上へクロック出力されると、「パケットの終り」(EOP)インタラプトが発生される。

【0113】8. ファームウェアは、別のフレームが送信の準備がなされているか否か(即ち、チェーン化されているか否か)をチェックする。

【0114】9. チェーン化されたフレームが存在する場合には、DMAコマンドが構築され且つ発行される。

【0115】10. 2番目のフレームの最初のバースト

が外部RAMからフェッチされ且つ送信FIFOメモリ内に書込まれる。

【0116】11. 書込バーストが終了し且つFIFOWMが超えられていない場合には、別のCOPが発行される。

【0117】12. この2番目のフレームに対する2番目のバーストを開始させるためにファームウェアは4番目のコマンドを構築する。

【0118】13. ファームウェアがCPOインタラプトをディスエーブルさせておらず且つFIFO内のデータがスタンダードのウォーターマークを超えていない場合には、別のCOPを発生することが可能である。

【0119】14. 該フレームの終端バイトがFIFOからネットワーク上にクロック出力されると、「パケットの終り」(EOP)インタラプトが発生される。

【0120】15. ファームウェアは、別のフレームが送信の準備がなされているか否か(即ち、チェーン化されているか否か)をチェックし、且つそうでない場合には、送信チャンネルをディスエーブルさせる。

【0121】LAWM取次フレーム送信(図24b)
1. DMAはパケットインタラプト信号の開始(SOP)を介してフレーム送信を開始させる。

【0122】2. ファームウェア(FW)は送信チャンネルをイネーブルさせ、コマンド(2個の32ビットワード)を構築し且つこのコマンドを実行のためにDMAへ提出する。

【0123】3. DMAはそのコマンドをデコードし、外部バスに対する調停を行い、外部共用メモリから適宜のデータを読み取り且つそれを適宜の送信FIFOメモリ内に書込む。LAWM信号が付加的なバーストに対してFIFO内に十分な容量が存在することを表わす場合には、ファームウェアはDMAに対して実行するために2番目のコマンドを提出する。

【0124】4. 各DMA転送が完了した後で且つ送信ウォーターマークが超えられていない場合には、パケットの継続(COP)インタラプトを発生することが可能である。

【0125】5. 該フレームの終端バイトがFIFOからネットワーク上にクロック出力されると、「パケットの終り」(EOP)インタラプトを発生することが可能である。

【0126】6. ファームウェアは、別のフレームが送信の準備がなされているか否か(即ち、チェーン化されているか否か)をチェックする。

【0127】7. チェーン化されたフレームが存在する場合には、DMAコマンドが構築され且つ発行される。

【0128】8. DMAはこの3番目のコマンドをデコードし、外部バスに対する調停を行い、外部共用メモリから適宜のデータを読み取り且つこれを適宜の送信FIFOメモリ内に書込む。LAWM信号が、FIFO内に付

加的なバーストに対する十分な容量が存在することを表わす場合には、ファームウェアは実行のためにDMAに対して4番目のコマンドを提出する。

【0129】9. 各DMA転送の後に送信ウォーターマークが超えられていない場合には、パケットの継続(COP)インタラプトを発生することが可能である。

【0130】10. 該フレームの終端バイトがFIFOからネットワーク上へクロック出力された場合に「パケットの終り」(EOP)インタラプトを発生することが可能である。

【0131】11. ファームウェアは別のフレームが送信の準備がされているか否か(即ち、チェーン化されているか否か)をチェックし、且つそうでない場合には、送信チャンネルをディスエーブルさせる。

【0132】ルックアヘッドウォーターマーク取次フレーム送信が有益なものであり且つ効率的であり且つ従来方法の場合に発生する待ち時間を解消していることが明らかである。

【0133】図25はパケット寸法に関連してインタラプト発生に関するウォーターマークの効果を示したグラフを示している。このグラフはFIFOウォーターマーク寸法の関数として発生されたインタラプトの数をプロットしている。このグラフから理解されるように、パケット寸法が増加すると、必要とされるインタラプトの数も増加する傾向にある。ウォーターマーク値は発生されたインタラプトの総数に関し逆の効果を有している。しばしばそうであるように、装置の性能をチューニングをする場合にウォーターマークのみの操作では不充分である。ネットワークパケット寸法の高い多様性及び共用システム資源に対する競合のために、付加的なメカニズムが望ましい。本発明のルックアヘッドウォーターマークはこのようなメカニズムであり且つそうであるから図25における曲線を押下げるものであることが容易に理解される。

【0134】III. 早期輻輳通知

本発明は、又、例えば受信FIFO70のうちの1つである対応するポート受信器内の輻輳のアドバンスト(先回り)ホスト通知に対するインタラプト(ECN)である早期輻輳通知信号を使用する。エラーを発生したフレームの前に先に受信したフレームがFIFO内に未だに格納されている場合があるので「アドバンスト」即ち先回りの用語を使用することが可能である。フレームの寸法及びFIFOの相対的な寸法に依存して、0からディスパッチ即ち送給されることを待機している多数のフレームの範囲のものが存在する可能性がある。従って、早期輻輳通知(ECN)の信号が最初に発生された時とエラーを発生したフレームが処理される時との間に著しく遅延が発生する場合がある。従来は、ホスト44は、その処理回路が先行するフレームの処理を行い且つそれがエラーを発生したフレームに来るまで全てのフレームの

ステータスワードを検査するまでこのタイプのエラーに気が付くものではなかった。ホストプロセッサ44はオーバーフロー問題に気が付くものではなかったので、その処理動作は修正されることなしに継続して進行し、従って、多数の先行するフレームは継続してFIFOをオーバーフローし従って喪失されるものであった。勿論、このことはより高いレベルのソフトウェアがフレームを再送することのより大きな要求を形成し、従って、ネットワークにおける帯域幅問題を形成していた。喪失フレーム問題を有する単一のダウンストリーム即ち下流側のノードの代わりに、その状態は、多数のダウンストリームノードがそれらの送信ウインドウを再クロック動作させることが強制されるものに迅速的に展開され、その問題を容易に悪化させるものであった。

【0135】本発明によれば、図26のフローチャートに示されるように、受信FIFOメモリにおけるネットワークデータ輻輳を制御する方法は、FIFOメモリ内のフレームオーバーフローを表わす受信FIFOメモリ内においてステータスエラーインジケータを発生するステップを有している(ブロック340)。早期輻輳インタラプトはステータスエラーインジケータにตอบสนองしてFIFOメモリから通信プロセッサに対して発生される(ブロック342)。そのインタラプトは処理され且つ少なくとも1個の早期輻輳通知ビットが直接メモリアccessユニットのマスタインタラプトレジスタ(MIR)内にセットされる(ブロック344)。

【0136】次いで、直接メモリアccessユニットからホストプロセッサに対して早期輻輳インタラプトが発生されて、FIFOメモリ内においてフレームオーバーフローが発生したことを表わす(ブロック346)。ホストプロセッサからFIFOメモリに対してフレームオーバーフローを発生させた入力フレームを廃棄するための命令が発生される(ブロック348)。受信フレームのサービスは直接メモリアccess(DMA)ユニットのバースト寸法のワード数を増加させるか、又はその他のアクティブなプロセスのタイムスライスを修正することのいずれか1つによって向上させることが可能である(ブロック350)。

【0137】図27A-Gは本発明の早期輻輳通知方法のハイレベルのブロック図を示している。図27Aは受信FIFOが空であり且つ読取(RD)及び書込(WR)ポインタが0, 0において同一であることを示している。次いで、データが入り始め且つ読取ポインタはゼロにあり且つ書込ポインタは図27Bに示したように前進する。パケットが受取られるに従い、ステータスがStat1によって表示される如くに書込まれる。2番目のフレーム即ちパケットが到着し(Data2)且つオーバーフローをし始める(図27C及び27D)。オーバーフロー条件が発生すると、エラーに対してフリップフロップがセットされ、従ってオーバーフロービットが

セットされる(図27G)。この点において、早期輻輳通知(ECN)が送り出される。書込ポインタはパケットの始めにリセットされ且つパケットの終りが発生するまで凍結され、その時に、低パケットのタイムエラーステータスフィールドへエンターする。DMAによるステータス1の読取がそれをホストアドレスにおける受信ステータスレジスタ内へコピーする。通信プロセッサがそのステータスを読取るまで、別のデータ転送のためのDMAの要求が発生することはない。このことはオーバーフローステータスによってステータスレジスタの上書きを防止する(図27E及び27F)。

【0138】次に、より特定的に図28-43を参照すると、3つの入って来る異なるパケットの場合について本発明方法及び装置について説明する。図28は受信FIFO72内においてデータが受信されていない場合のネットワーク制御器及びホストシステムを示している。図29において、データは、最初に、受信FIFO72へ入り、且つ図30においてウォーターマーク258に到達し且つパケットの始めインタラプトがインタラプトバス252を介して通信プロセッサ92へ送られる。通信プロセッサ92はデータを転送するためにDMA85に対してコマンドを発行する(図31)。同時に、データは矢印で示されるように継続して受信FIFO72へエンターする。

【0139】図32に示したように、DMAはシステムバス42の所有についてバスアービトレーション(調停)論理ユニット47と共にネゴシエーションを行い、一方データは継続して受信FIFOメモリ72内へ転送される。図33において、DMA85は受信FIFO72からのデータを共用システムメモリ46へ転送する。図34に示したように、2番目のパケット即ちフレームが受信FIFOメモリ72へ入る。図35、36、37は、システムバス42に対するアクセスが拒否されたという点を除いて、図30、31、32と同様である。この時に、3番目のパケット(黒塗り陰影)が2番目のパケット(対角線陰影)と共にエンターする(図38)。図39において、入って来るフレームが受信FIFOメモリ72をオーバーフローし且つ早期輻輳通知(ECN)ビットがセットされた後に(図27G)内部インタラプトが通信プロセッサ92へ送られる。図41において、通信プロセッサ92はDMA85の適宜のレジスタブロック内のポートに対するECNビットをセットする。図42において、DMA85はシステムバス42に沿ってホストプロセッサ44に対し早期輻輳インタラプトの信号を送り且つDMA85は受信FIFO72からのデータを図43に示したように共用システムメモリ46へ転送する。3番目のフレームは失われる。然しながら、上のレベルのソフトウェアはそのフレームを送信することが可能である。

【0140】IV. フェンスポスト(Fence Post

st)

再度図3、4、5、7を参照して、記述子リング202及び記述子206の説明に関して更に詳細に検討する。図44のグラフに加えて、本発明方法及び装置はホスト44とネットワーク制御器40との間でフレームに調整されたデータの転送を制御することが明らかである。本発明によれば、バスの利用を向上させ且つ最初の記述子及び最後の記述子及び中間の記述子の所有を所望のホスト又は制御器へ許可するために、記述子「チェーン」内における最初及び最後の記述子206のみがアップデートされる。

【0141】前述したように、ホスト44は受信されたか又は送信されたフレームよりも寸法がより小さなフレームデータバッファ204を使用すべく選択することが可能であり、従って、単一のフレームデータバッファは複数のフレームデータバッファ204をスパンする、即ちそれらのバッファにわたることが可能である。このことはフレームをネットワーク制御器40によって切断するか又は組立てることを可能とする。当然、上述したように、複数のフレームデータバッファ204は関連する記述子206と一緒に「チェーン化」することによってフレームを構成する断片及びそのチェーンの最後の記述子内にフレームの終りのフラグがセットされている記述子リング202内の連続するエントリを保持することが可能である。所有されているがそのフレームの終りフラグがセットされていない記述子エントリ206の夫々のフレームデータバッファはフレームの一部であり全体的なフレームではないと考えられる。制御器40は、それが各相次ぐフレームデータバッファ204を充填する場合に1つづつ記述子206と一緒にチェーン化することが可能である。フレームの終りが最終的に受取られ且つ外部共用メモリ46へ転送されると、そのフレームの終りフラグが記述子チェーンの最後の記述子においてセットされる(図4)。

【0142】送信期間中に、制御器40は単一のフレーム及び当然に「チェーン化」された記述子206によってポイントされる「チェーン化」されたフレームデータバッファ204の内容を逐次的に構築することが可能である。フレームの送信は、その記述子206がフレームの終りフラグをセットしているフレームデータバッファ204に遭遇する場合にのみ終了する。このバス利用における著しい改良は、本発明によって発生され、その場合に、各スパンされている記述子206を逐次的にアップデートする従来技術の代わりに、例えば、ネットワークが受信したフレームに対する記述子内の所有ビットをアップデートすることによって、最初の記述子と最後の記述子のみを変更させる。これらのアップデートされた最初の記述子及び最後の記述子はチェーンの「フェンスポスト」即ち垣根の支柱を形成する。

【0143】「フェンスポスト」型チェーン内の最初及

び最後の記述子の全てのフラグ及びフィールドは、完全に送信されるか又は受信されるとフレームに関する正確な情報を提供するために、アップデートされる。例えば、受信フレームの場合には、該チェーン内の最初の記述子のメッセージ寸法フィールド218が、単にバッファ寸法に等しい関連するバッファのバイトカウントではなく、全体的なフレームのバイトカウントでアップデートされる。

【0144】上述したように、図4はチップ初期化セクション200a及び統計イメージ200b-eを具備する4個のポートを具備する管理ブロック200を示している。記述子リング202は、アドレスを使用してフレームデータバッファに対してポイントする種々の記述子206と共に示されている。フレームデータバッファは右側に示してある。図5は2ワードエントリとしての記述子26と、所有ビット(OB)212及びパケットの終り(EOP)214を有するフレームデータバッファ204を示している。バッファ寸法216及びメッセージ寸法218は1つのワード208内に収容されており、且つバッファアドレス219は他のワード210内に収容されている。図44におけるグラフは、上述した如く最初の記述子及び最後の記述子のみを使用することが平坦なラインを形成しパスに沿ってのトラフィックを減少させることを詳細に示している。

【0145】図3は、更に、どのようにして管理ブロック200が例えば幾何学的形状等のバッファ情報206a及びバッファアドレス206bを具備する記述子206を持った異なる送信リング202に対して直接的にポイントしているポインタ200d(図7)を有しているかを示している。

【0146】V. 記述子リングの形成

本発明は、ネットワーク装置が、例えば記述子リング等のデータ及びバッファ構造を形成する役目を担っているもので有益である。ネットワーク装置40は外部共用メモリ46において送信及び／又は受信記述子リング202(図3)を構築する。本発明においては、全二重チャンネルに対するサポートが提供されている。送信又は受信記述子リング202のいずれかにおける記述子の数を支配するパラメータ及びそれらの夫々のフレームデータバッファ寸法はパラメータブロック(又は管理ブロック)を介して通信される。

【0147】この管理ブロック200はホスト制御下における通信プリミティブを介して初期化(図45)においてホストシステム43とネットワーク装置40との間で交換される。管理ブロック200はメモリ46の多数の可変フィールド内に格納される(即ちマッピングされる)。上述したように、送信記述子リング寸法又は受信記述子リング寸法に対するフィールド値が非ゼロである場合には、構築を開始することが可能である。そうではなく、フィールドがゼロである場合には、ネットワーク

装置40は関連する記述子リング202を構築することはない。ネットワーク装置40は、ホスト40が共用メモリ46内に既にデータ及びメモリ構造を構築したものと期待する。記述子リング202の幾何学的形状即ち長さ及び関連するフレームデータバッファ204の寸法は異なり且つ記述子リング202は、しばしば、50個から500個の記述子の長さが変化し、一方フレームデータバッファ204は約256個のバイトから最大で約2,000又は5,000個のバイトに変化する。フレームデータバッファ寸法は、インターフェースネットワークの最大のサポートされているフレーム寸法に基づいて選択される。ポート50-56毎に割当てられている全体的なメモリは2メガバイト範囲内である。フレームデータバッファ寸法は記述子リング202を実際に構築するために必要な時間に比較的殆ど影響を有するものではない。然しながら、記述子リング寸法は構築時間に対する制限的要因である。ブロックモード構築最適化技術を使用して構築時間を減少させる。記述子206は2つのブロック内でオンチップで構築し且つ直接メモリアクセスユニット85を介して外部メモリ46へ転送することが可能である。

【0148】このブロック寸法は変更可能であり且つ将来ブロックのパラメータ内に容易に包含させることが可能である。本発明の方法及びネットワーク装置は種々の利点を有しており、例えばホストソフトウェアの開発に対して必要な時間が減少されており且つホストコードの寸法が減少されている。テストを促進させ且つより高速のネットワーク装置初期化とすることが可能である。又、本発明はアプリケーション設計エンジニアに対するシステムの実現を促進させる。

【0149】本発明によれば、共用メモリ46内のメモリのブロックがホストシステム43によって割当てられ、それは上述したような記述子リングパラメータ200bを有する管理ブロック200をマッピングさせる(図7)。これらのパラメータは、共用メモリ内において形成されるべき記述子リング202及び記述子204の幾何学的形状を包含している。図7は管理ブロックを示しており且つ4つのアドレスPAD+60乃至PAD+72において、バッファ寸法、送信リング寸法及び受信リング寸法を表わしている。

【0150】図45に示したように、管理ブロック200はチャート上の点0においてセットアップされたベースポインタを有している。ホストシステム43はネットワーク装置に対して初期化用のプリミティブ(点1におけるINIT)を発行する。同時に、ホスト44は管理ブロック200のベースアドレスをネットワーク装置40内に書込む。ネットワーク装置40は共用メモリ(点2)から管理ブロックを「フェッチ」即ち読取り、次いで、管理ブロックが受取られたことのアクノジメント(ACK)をホストへ送る。この管理ブロックが処理さ

れ、一方ホストシステムはアクノレジメントを受取った後に付加的なハウスキーピング（点 3）を行うことが可能である。

【0 1 5 1】管理ブロック 2 0 0 が初期されるに従いネットワーク装置 4 0 は共用メモリ内に形成されるべきフレームデータバッファに対してポイントするデータのブロックとして対応する記述子を構築する。

【0 1 5 2】図 4 6 は、どのようにして記述子がネットワーク装置によって形成することが可能であるかを示したフローチャートを詳細に示している。ブロック 4 0 0 において、ホストはベース記述子リング及び関連するバッファに対してポイントを提供する。前述したように、送信リング寸法又は受信リング寸法フィールドに対するフィールド値が非ゼロである場合には、構築がすぐさま開始される。そうでなく、これらのフィールドがゼロである場合には、ネットワーク装置は関連する記述子リングを構築することではなく、ホストが共用メモリ内に既にそのような構造を構築したものと期待する。

【0 1 5 3】管理ブロックはネットワーク装置によって読取られ（ブロック 4 0 2）且つ記述子ヘッダワードが構築される（ブロック 4 0 4）。記述子アドレスワードが構築され（ブロック 4 0 6）且つ記述子アドレスがアップデートされる（ブロック 4 0 8）。バッファポイントアドレスもアップデートされ（ブロック 4 1 0）且つ、記述子ブロックがネットワーク装置によって共用システムメモリの一部であるホスト RAM に対して読み出される（ブロック 4 1 2）。

【0 1 5 4】次いで、ホストは、それが完了しているか否かを判別するためにテストされ（ブロック 4 1 4）、且つ、完了していない場合には、記述アドレスが再度アップデートされる。その処理が完了していると、E O R ビットが終端記述子に対してセットされ（ブロック 4 1 6）且つ終端記述子はホスト（R A N）へ書き出される（ブロック 4 1 8）。次いで、このプロセスは終了する（ブロック 4 2 0）。例えば隣接した記述子の使用、及びイベントカウント等の多数の仮定が存在している。典型的に、バッファは隣接したもの即ち連続的なものであり且つ一様な寸法のものである。バッファポイントが提供されない場合には、ファームウェア 1 0 2 は記述子リングの計算された終端から 2 ワードオフセットされているバッファを開始させる。管理ブロック記述子パラメータ 1 6 進数ワードが「0 X 0 0 0 0 0 0 0 0」である場合には、関連する記述子リング 2 0 2 が構築されることはない。管理ブロック転送は、その他のコンフィギュレーション（形態特定）プリミティブの前に必要とされる。何故ならば、そのブロックはその設定を上書きするからである。全ての記述子リング寸法は偶数値でなければならず且つフレームデータバッファ寸法は 0 又は 1 とすることが可能であり又は記述子リング 2 0 2 は構築されない。全てのバッファポイントはリング寸法に拘らず

に強制的にアライメント即ち整合が与えられる。構築することの可能な最も小さな記述子リングは寸法において 3 個の記述子であり、且つ DMA 転送当たり 1 個のブロックでブロック当たり 2 個の記述子である。

【0 1 5 5】図 4 7 - 5 0 は使用することの可能な種々のフィールド及びビット値のみならず送信及び受信メッセージ記述子の更なる詳細を示した表を例示している。

【0 1 5 6】本発明に関連する技術的事項は、本願と同日付をもって出願される本願出願人が出願人である以下の発明の名称を有する特許出願に記載されており、これらの各出願の開示内容も引用により本願明細書に取込むこととする。

【0 1 5 7】1. 「フレームアドレス通知を使用してネットワークをベースとしたデータのルーチング方法及びシステム」（整理番号 S T 6 3 1）

2. 「F I F O メモリ内への付加的なデータバースト用のルックアヘッドウォーターマーク」（整理番号 S T 6 3 2）

3. 「ネットワークデータ輻輳を制御する方法及び装置」（整理番号 S T 6 3 3）

4. 「共用メモリ内にバッファ構造を形成する方法及びネットワーク装置」（整理番号 S T 6 3 5）

以上、本発明の具体的実施の態様について詳細に説明したが、本発明は、これら具体例にのみ制限されるべきものではなく、本発明の技術的範囲を逸脱することなしに種々の変形が可能であることは勿論である。

【図面の簡単な説明】

【図 1】 3 2 ビットシステムバスへ接続する本発明のネットワーク制御器として示してあり且つホストシステムマイクロプロセサと、バスアービトレイション（調停）論理ユニットと、共用メモリサブシステムとを示している 4 個のネットワーク装置のハイレベルブロック図。

【図 2】 本発明のネットワーク制御器を示しており且つ 4 個のポートと、通信プロセサと、システムバスインターフェース制御ユニットとを示しているハイレベルのブロック図。

【図 3】 本発明の装置及びネットワーク制御器によって使用されるバッファ管理及びシステムメモリを示しており且つ種々の記述子リングを示しているハイレベルのブロック図。

【図 4】 管理ブロックと、記述子リングと、フレームデータバッファとを示したデータ構造及びシステムメモリのハイレベルのブロック図。

【図 5】 記述子及びバッファのハイレベルのブロック図。

【図 6】 本発明のネットワーク制御器のタイマ動作のハイレベルのブロック図。

【図 7】 本発明において使用されているシステムメモリ及び管理ブロックの詳細を示した概略図。

【図8】 本発明のシステムメモリ及び統計イメージ及び管理ブロックのブロック図及びチャートを示した概略図。

【図8A】 本発明において使用されている直接メモリアクセスユニットのプリミティブコマンドレジスタに対する種々のビット値及び記述を示した表。

【図8B】 本発明において使用されている直接メモリアクセスユニットのマスタインタラプトレジスタに対する種々のビット値及び記述を示した表。

【図9】 レイヤ構造の1例としての種々のヘッダの階層的コンフィギュレーション即ち形態を示したブロック図。

【図10】 802.3データリンクレイヤヘッダを示したブロック図。

【図11】 インターネットIPヘッダを示したブロック図。

【図12】 TCPヘッダを示したブロック図。

【図13】 ネットワーク制御器及び外部ホストプロセサ、バスアービトリション論理ユニット及び共用システムメモリの基本的なコンポーネントを示しており且つ本発明のフレームアドレス通知に対するシーケンスにおけるあるステップにおける状態を詳細に示したハイレベルのブロック図。

【図14】 ネットワーク制御器及び外部ホストプロセサ、バスアービトリション論理ユニット及び共用システムメモリの基本的なコンポーネントを示しており且つ本発明のフレームアドレス通知に対するシーケンスにおけるあるステップにおける状態を詳細に示したハイレベルのブロック図。

【図15】 ネットワーク制御器及び外部ホストプロセサ、バスアービトリション論理ユニット及び共用システムメモリの基本的なコンポーネントを示しており且つ本発明のフレームアドレス通知に対するシーケンスにおけるあるステップにおける状態を詳細に示したハイレベルのブロック図。

【図16】 ネットワーク制御器及び外部ホストプロセサ、バスアービトリション論理ユニット及び共用システムメモリの基本的なコンポーネントを示しており且つ本発明のフレームアドレス通知に対するシーケンスにおけるあるステップにおける状態を詳細に示したハイレベルのブロック図。

【図17】 ネットワーク制御器及び外部ホストプロセサ、バスアービトリション論理ユニット及び共用システムメモリの基本的なコンポーネントを示しており且つ本発明のフレームアドレス通知に対するシーケンスにおけるあるステップにおける状態を詳細に示したハイレベルのブロック図。

【図18】 ネットワーク制御器及び外部ホストプロセサ、バスアービトリション論理ユニット及び共用システムメモリの基本的なコンポーネントを示しており且つ

本発明のフレームアドレス通知に対するシーケンスにおけるあるステップにおける状態を詳細に示したハイレベルのブロック図。

【図19】 ネットワーク制御器及び外部ホストプロセサ、バスアービトリション論理ユニット及び共用システムメモリの基本的なコンポーネントを示しており且つ本発明のフレームアドレス通知に対するシーケンスにおけるあるステップにおける状態を詳細に示したハイレベルのブロック図。

【図20】 ネットワーク制御器及び外部ホストプロセサ、バスアービトリション論理ユニット及び共用システムメモリの基本的なコンポーネントを示しており且つ本発明のフレームアドレス通知に対するシーケンスにおけるあるステップにおける状態を詳細に示したハイレベルのブロック図。

【図21】 本発明のフレームアドレス通知の送信インタラプトイベントタイムラインを一般的に示したタイミング線図。

【図22】 本発明のルックアヘッドウォーターマークを使用した流れ制御とクラシックな先入先出流れ制御との比較を示したブロック図。

【図23】 本発明のルックアヘッドウォーターマークを使用したプロセスを示したフローチャート。

【図24A】 インタラプト取次型フレーム送信を示したタイミング線図。

【図24B】 ルックアヘッドウォーターマーク取次型フレーム送信を示したタイミング線図。

【図25】 ウォーターマーク値がどのようにして発生されたインタラプトの総数に対して逆の効果を有しているかを示したグラフ図。

【図26】 本発明の早期輻輳通知信号を使用する基本的なプロセスを示したフローチャート。

【図27】 (A)乃至(G)はどのようにして先入先出メモリが受信FIFOメモリ内への2番目のパケット上でオーバーフローするか及び種々の読取及び書込ステータスポインタを示したハイレベルブロック図。

【図28】 本発明のネットワーク制御器の外部ホストプロセサ、バスアービトリション論理ユニット、共用メモリ、基本コンポーネントを示しており且つ早期輻輳通知信号が3番目のパケットに関するオーバーフローを有する3つの異なる入力パケットに対して使用される場合のプロセスの1つの段階における状態を示したハイレベルブロック図。

【図29】 本発明のネットワーク制御器の外部ホストプロセサ、バスアービトリション論理ユニット、共用メモリ、基本コンポーネントを示しており且つ早期輻輳通知信号が3番目のパケットに関するオーバーフローを有する3つの異なる入力パケットに対して使用される場合のプロセスの1つの段階における状態を示したハイレベルブロック図。

プロセサ、バスアービトリエーション論理ユニット、共用メモリ、基本コンポーネントを示しており且つ早期輻輳通知信号が3番目のパケットに関するオーバーフローを有する3つの異なる入力パケットに対して使用される場合のプロセスの1つの段階における状態を示したハイレベルブロック図。

【図38】 本発明のネットワーク制御器の外部ホストプロセサ、バスアービトリション論理ユニット、共用メモリ、基本コンポーネントを示しており且つ早期輻轉通知信号が3番目のパケットに関するオーバーフローを有する3つの異なる入力パケットに対して使用される場合のプロセスの1つの段階における状態を示したハイレベルブロック図。

【図39】 本発明のネットワーク制御器の外部ホストプロセサ、バスアービトリション論理ユニット、共用メモリ、基本コンポーネントを示しており且つ早期輻輳通知信号が3番目のパケットに関するオーバーフローを有する3つの異なる入力パケットに対して使用される場合のプロセスの1つの段階における状態を示したハイレベルブロック図。

【図４０】 本発明のネットワーク制御器の外部ホストプロセサ、バスアービトラクション論理ユニット、共用メモリ、基本コンポーネントを示しており且つ早期輻輳通知信号が３番目のパケットに関するオーバーフローを有する３つの異なる入力パケットに対して使用される場合のプロセスの１つの段階における状態を示したハイレベルブロック図。

【図４１】 本発明のネットワーク制御器の外部ホスト
プロセサ、バスアービトレーション論理ユニット、共用
メモリ、基本コンポーネントを示しており且つ早期輻輳
通知信号が３番目のパケットに関するオーバーフローを
有する３つの異なる入力パケットに対して使用される場
合のプロセスの１つの段階における状態を示したハイレ
ベルブロック図。

【図４２】 本発明のネットワーク制御器の外部ホスト
プロセサ、バスアービトリション論理ユニット、共用
メモリ、基本コンポーネントを示しており且つ早期輻輳
通知信号が３番目のパケットに関するオーバーフローを
有する３つの異なる入力パケットに対して使用される場
合のプロセスの１つの段階における状態を示したハイレ
ベルブロック図。

【図４３】 本発明のネットワーク制御器の外部ホストプロセサ、バスアービトリション論理ユニット、共用メモリ、基本コンポーネントを示しており且つ早期輻輳通知信号が３番目のパケットに関するオーバーフローを有する３つの異なる入力パケットに対して使用される場合のプロセスの１つの段階における状態を示したハイレベルブロック図。

【図 4 4】 最初の記述子と最後の記述子のみがアップデートされる場合の通常の記述子及び「フェンスポス

ト」を使用したホストバスの推測されるトラフィック構成を詳細に示したグラフ図。

【図45】 ホストシステムと本発明のネットワーク装置、例えばネットワーク制御器との間でのプリミティブ（基本命令）の信号処理を示したチャート。

【図46】 ネットワーク装置内での記述子を構築するプロセスを示したフローチャート。

【図47】 受信及び送信メッセージ記述子の種々のフィールドを示したテーブル。

【図48】 受信及び送信メッセージ記述子の種々のフ 10

ィールドを示したテーブル。

【図49】 受信及び送信メッセージ記述子の種々のフ

ィールドを示したテーブル。

【図50】 受信及び送信メッセージ記述子の種々のフ

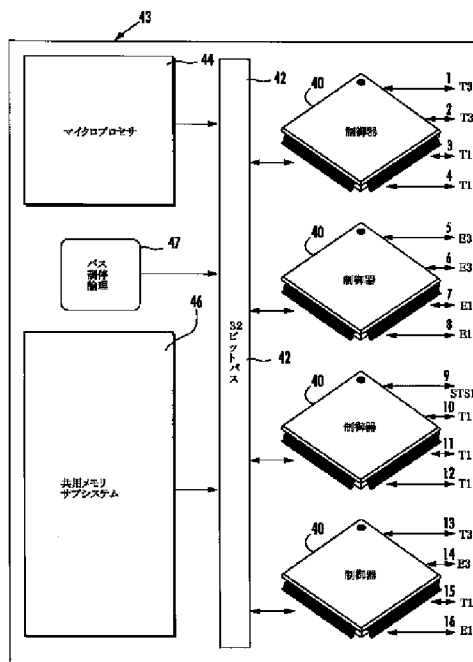
ィールドを示したテーブル。

【符号の説明】

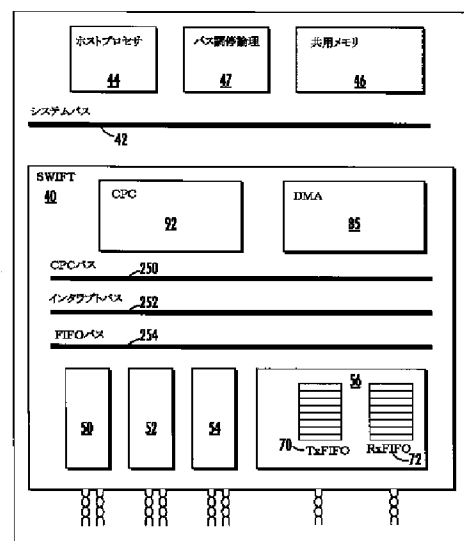
- 40 ネットワーク制御器
- 42 システムバス
- 43 ホストバス
- 44 ホストマイクロプロセサ
- 46 共用メモリサブシステム
- 50, 52, 54, 56 ポート
- 58 全二重プロトコル線

- 60 送信データハンドラ
- 62 ライントランシーバ
- 64 受信データハンドラ
- 66, 68 FIFO論理回路
- 70 送信FIFO
- 72 受信FIFO
- 74 制御回路
- 78 管理バス
- 80 インターフェース制御ユニット（SBI）
- 82 共用バスインターフェース回路
- 84 バススレーブ制御器
- 85 直接メモリアクセスユニット
- 86 コンフィギュレーションデータ転送エンジン
- 88 管理データ転送エンジン
- 90 フレームデータ転送エンジン
- 92 通信プロセサ（CPC）
- 94 CPU
- 96 ALU
- 98 タイマ
- 100 RAM
- 102 ファームウェアROM
- 104 インタラプトハンドラ

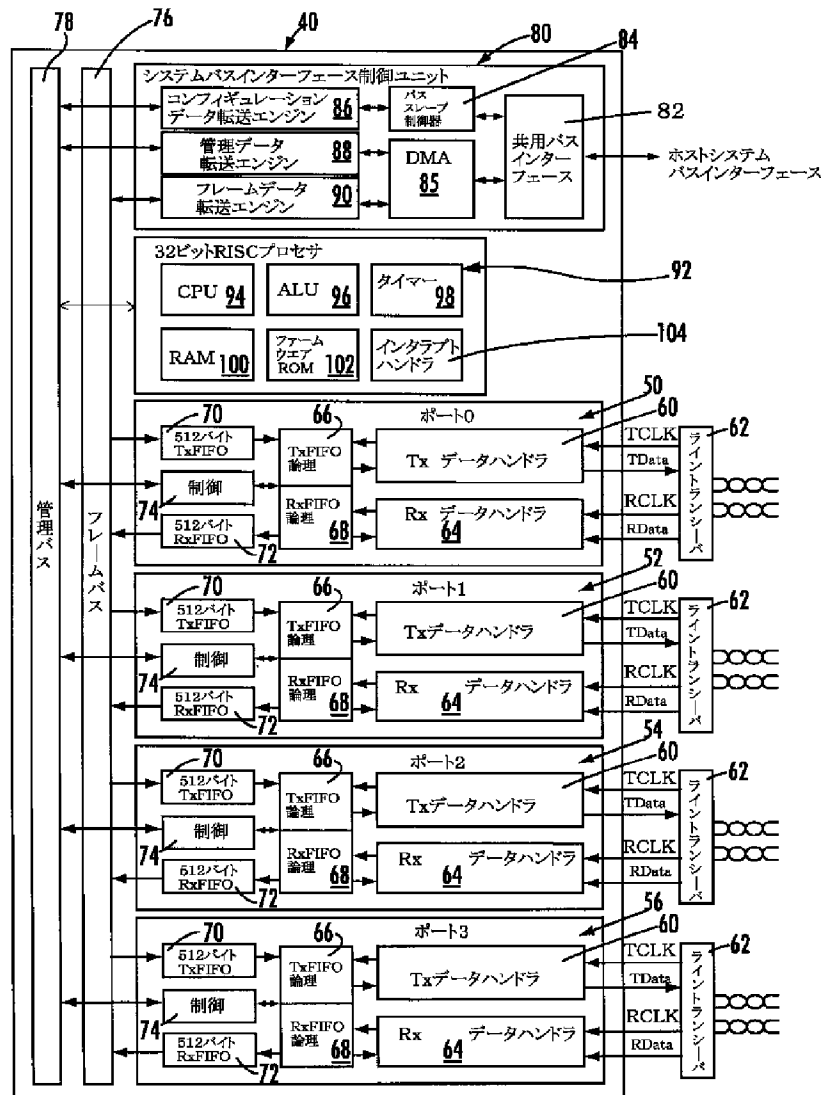
【図1】



【図13】



【図2】

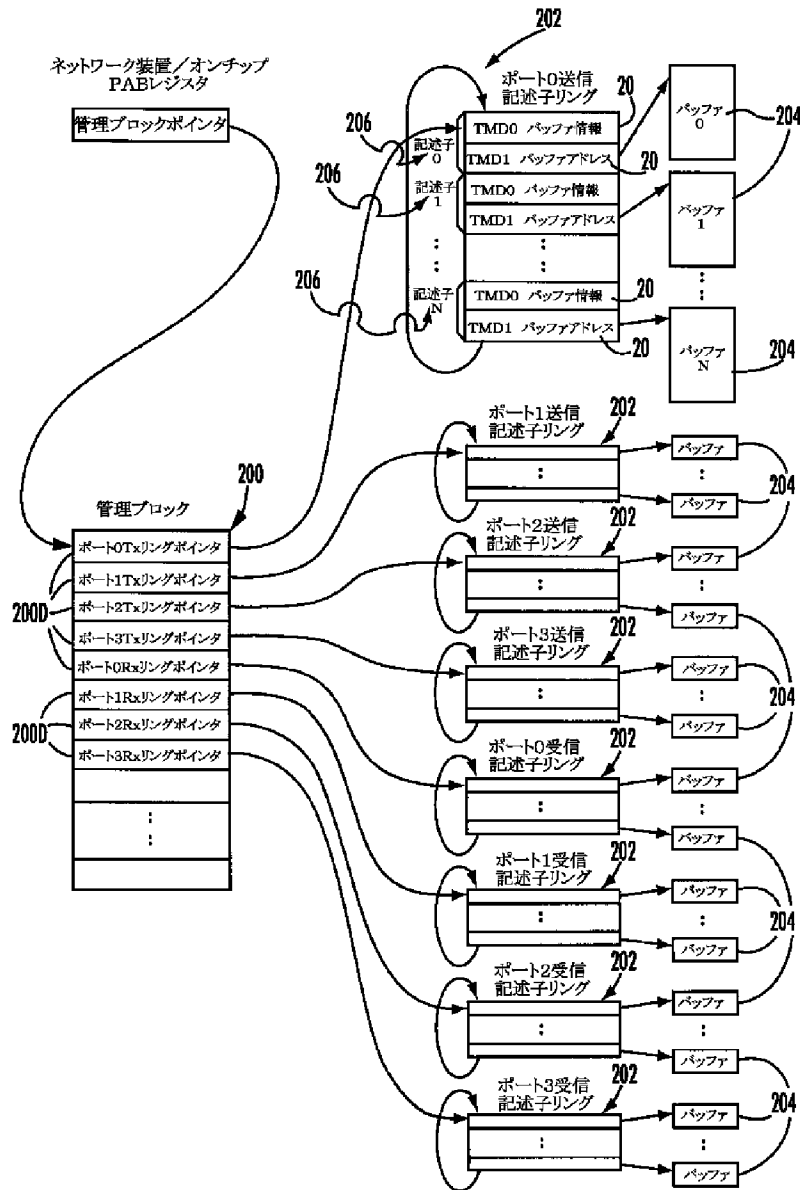


【図11】

インターネットIPヘッダ(20バイト)

VER/ヘッダ	サービスのタイプ	16ビット全長(バイト単位)
16ビットID		3ビットフラグ/13ビット断片オフセット
TTL	8ビットプロトコル	16ビットヘッダチェックサム
32ビット発信元IPアドレス		
32ビットデスティネーションIPアドレス		
(オプション-ある場合)		

【図3】

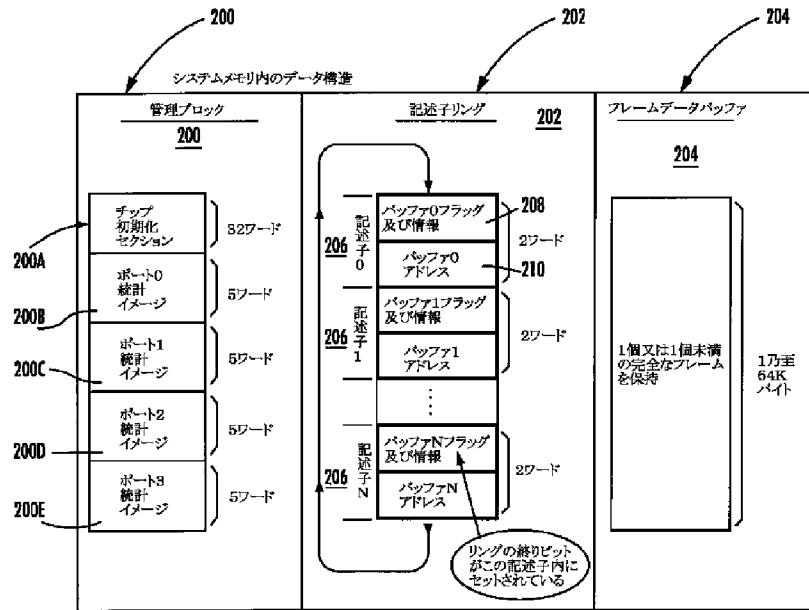


【図12】

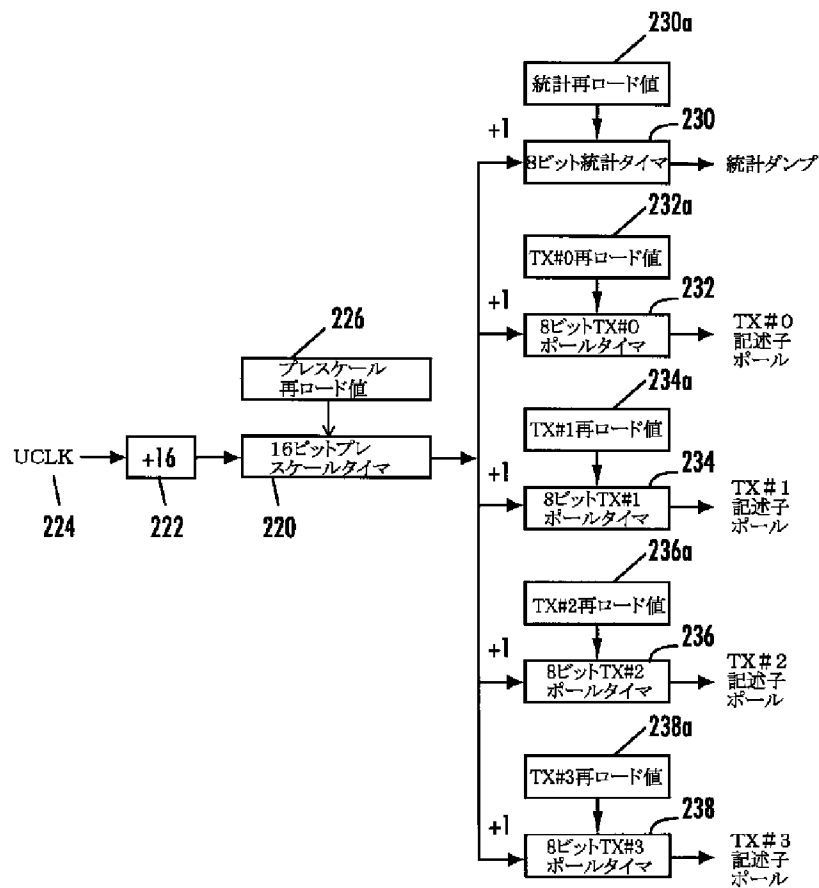
TCPヘッダ(20バイト)

16ビット発信元ポート	16ビットデスティネーションポート
32ビットシーケンス番号	
32ビットアキュレジメント番号	
URG/ACK/PSH/RST/SYN/FIN	16ビットウィンドウ寸法
16ビットTCPチェックサム	16ビットアージェントポインタ

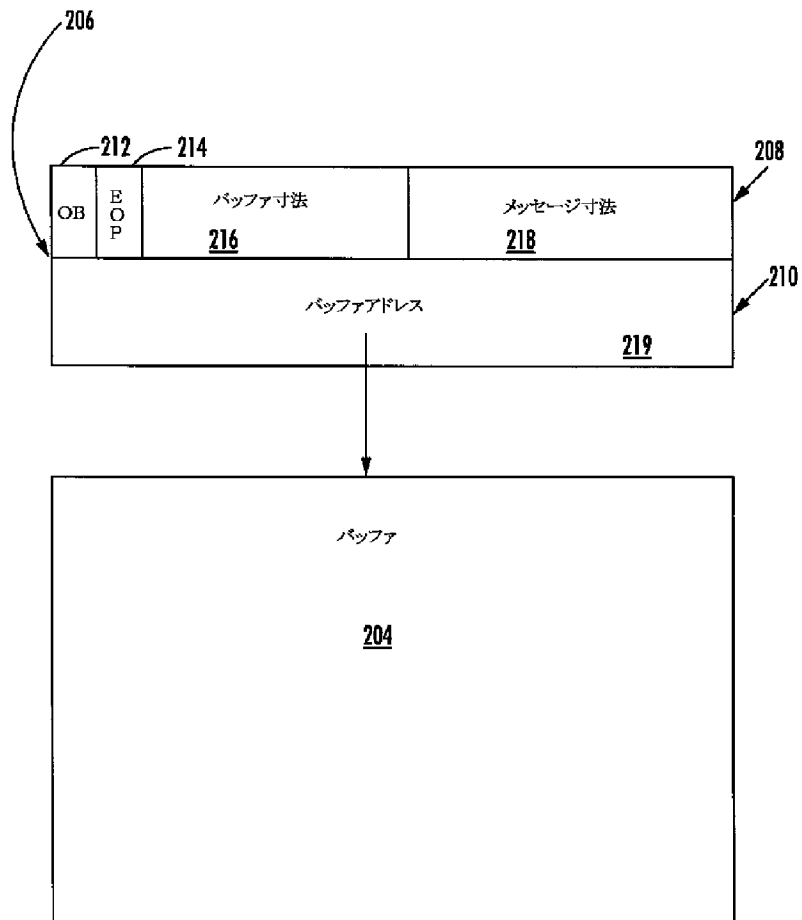
【図4】



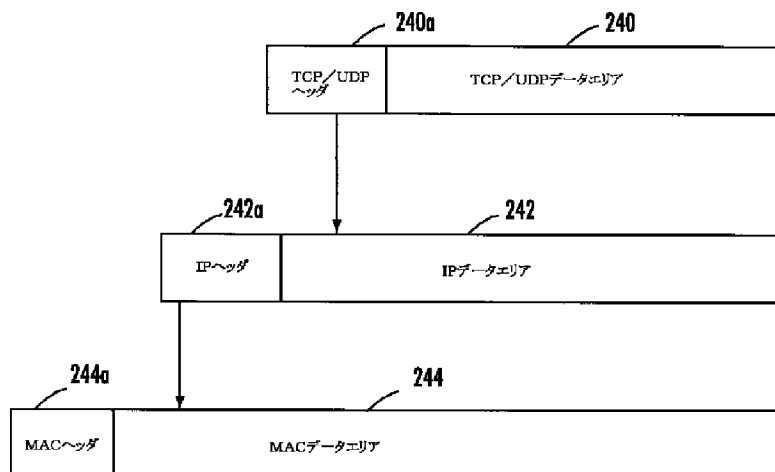
【図6】



【図5】



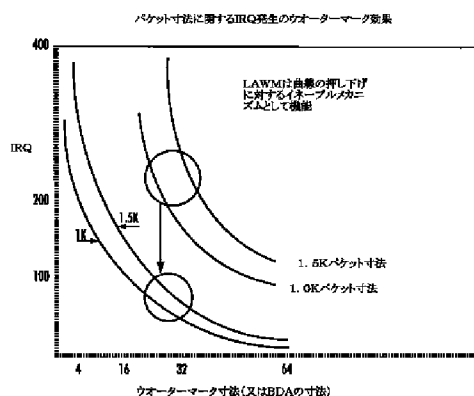
【図9】



【図7】

相対 アドレス	バイト3	バイト2	バイト1	バイト0		
PAB+0	ポート0 Tx リングトップ記述子ポインタ				200D	
PAB+4	ポート1 Tx リングトップ記述子ポインタ					
PAB+8	ポート2 Tx リングトップ記述子ポインタ					
PAB+12	ポート3 Tx リングトップ記述子ポインタ					
PAB+16	ポート0 Rx リングトップ記述子ポインタ					
PAB+20	ポート1 Rx リングトップ記述子ポインタ					
PAB+24	ポート2 Rx リングトップ記述子ポインタ					
PAB+28	ポート3 Rx リングトップ記述子ポインタ					
200A	PAB+32	プレスケールタイム 再ロード値		統計タイム 再ロード値	タイム イネーブル	200E
	PAB+36	ポート3Tx ボールタイム 再ロード値	ポート2Tx ボールタイム 再ロード値	ポート1Tx ボールタイム 再ロード値	ポート0Tx ボールタイム 再ロード値	
200B	PAB+40	ポート3Tx バースト寸法	ポート2Tx バースト寸法	ポート1Tx バースト寸法	ポート0Tx バースト寸法	200F
	PAB+44	ポート3Rx バースト寸法	ポート2Rx バースト寸法	ポート1Rx バースト寸法	ポート0Rx バースト寸法	
	PAB+48	予約済み	予約済み	UCLK期間 (ナノ秒)	統計バースト寸法	
200C	PAB+52	ポート1 N1		ポート0 N1		200G
	PAB+56	ポート3 N1		ポート2 N1		
	PAB+60	ポート#0パッファ寸法		Txリング寸法	Rxリング寸法	200H
	PAB+64	ポート#1パッファ寸法		Txリング寸法	Rxリング寸法	
	PAB+68	ポート#2パッファ寸法		Txリング寸法	Rxリング寸法	200I
	PAB+72	ポート#3パッファ寸法		Txリング寸法	Rxリング寸法	
	PAB+76	予約済み				200J
	PAB+80	予約済み				
PAB+84	予約済み					
PAB+88	予約済み					
PAB+92	予約済み					
PAB+96	予約済み					
PAB+100	予約済み					
PAB+104	予約済み					
PAB+108	予約済み					
PAB+112	予約済み					
PAB+116	予約済み					
PAB+120	予約済み					
PAB+124	予約済み					

【図25】



【図8】

ポート0 相対アドレス	バイト3	バイト2	バイト1	バイト0	ポート
PAB+128	不良フレーム受信				ポート#0
PAB+132	アボートしたフレーム				
PAB+136	N1を超えて受信したフレーム				
PAB+140	予約済み				
PAB+144	予約済み				
PAB+148	不良フレーム受信				ポート#1
PAB+152	アボートしたフレーム				
PAB+156	N1を超えて受信したフレーム				
PAB+160	予約済み				
PAB+164	予約済み				
PAB+168	不良フレーム受信				ポート#2
PAB+172	アボートしたフレーム				
PAB+176	N1を超えて受信したフレーム				
PAB+180	予約済み				
PAB+184	予約済み				
PAB+188	不良フレーム受信				ポート#3
PAB+192	アボートしたフレーム				
PAB+196	N1を超えて受信したフレーム				
PAB+200	予約済み				
PAB+204	予約済み				

【図8A】

[0x28]PCR-プリミティブコマンドレジスタ

DMA	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名称	PPA								PPARM[7:0]								HPRIM[6:0]								HPARM[7:0]							
リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ホスト アクセス	リードオンリ								リードオンリ								読取／書込								読取／書込							
CPC アクセス	読取／書込								読取／書込								リードオンリ								リードオンリ							

ビット#	フィールド	名称	記述
31	PPA	プロバイダ プリミティブ 使用可能	(1=使用可能;0=プリミティブ無し)ファームウェアによりPCRレジスタが書込まれる場合に装置によって設定。このビットの設定はMIRのPINTビットも自動的に設定させる。このビットはホストがこのレジスタを読取る場合にDMAによってクリアされる。
30:24	PPRIM	プロバイダ プリミティブ コマンド	(7ビット二進値)このフィールドは外へ行く(ファームウェアからホストへ)プリミティブコマンドである。その意味は厳格にファームウェアによって決定される。
23:16	PPARM	プロバイダ プリミティブ パラメータ	(8ビット二進値)これはプロバイダプリミティブコマンドに対応するファームウェアが定義したパラメータフィールドである。
15	HPA	ホスト プリミティブ 使用可能	(1=使用可能;0=プリミティブ無し)ホストによりPCRレジスタが書込まれる場合に装置により設定される。このビットの設定はイネーブルされた場合にCPCインタラプトとなる場合がある。このビットはファームウェアがこのレジスタを読取る場合にDMAによりクリアされる。
14:8	HPRIM	ホスト プリミティブ コマンド	(7ビット二進値)このフィールドは入ってくる(ホストからファームウェアへ)プリミティブコマンドである。その意味は厳格にファームウェアによって決定される。
7:0	HPARM	ホスト プリミティブ パラメータ	(8ビット二進値)これはホストプリミティブコマンドに対応するファームウェアが定義したパラメータフィールドである。

【図8B】

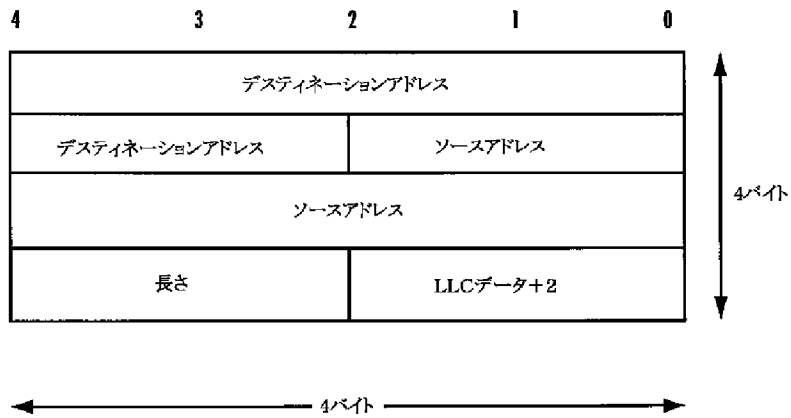
[0x2A]MIR-マスタインタラプトレジスタ

DMA	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名称	PINT	SPURINT	SDRIFT3	ECN3	FAN3	SHL3	TINT3	RINT3	MERR	PPLOST	SDRIFT2	ECN2	FAN2	SHL2	TINT2	RINT2	SERR	HPLOST	SDRIFT1	ECN1	FAN1	SHL1	TINT1	RINT1	WERR	SPARE	SDRIFT0	ECN0	FAN0	SHL0	TINT0	RINT0
リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ホストアクセス	* 読取り—クリア (書込無し)																															
CPC アクセス	読取り 0	読取り—0/ 書込—1							読取り 0	読取り—0/ 書込—1							読取り 0	読取り—0/ 書込—1							読取り 0	読取り—0/ 書込—1						

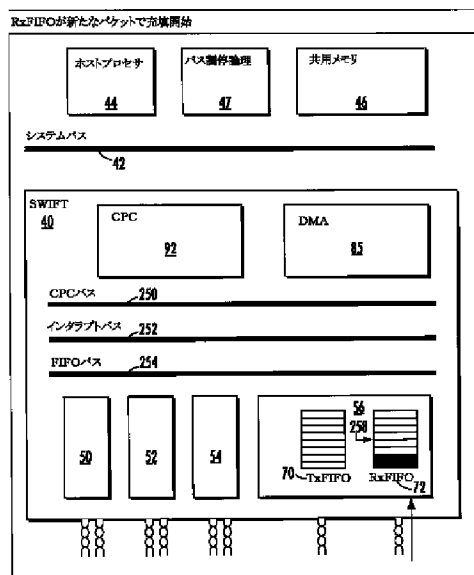
ビット#	フィールド	名称	記述
31	PINT	プリミティブ インタラプト	(1=イベント;0=イベント無し)ファームウェアがプリミティブコマンドレジスタ (上半分)内に新たなプロバイダプリミティブを書込む場合に装置によって設定
23	MERR	メモリ エラー	(1=イベント;0=イベント無し)システムモードレジスタ(SMR)において定義され 且つ確立されるようにRTIME/システムタイムアウトが発生した場合に装置によって設定
15	SERR	システム エラー	(1=イベント;0=イベント無し)システムモードレジスタ(SMR)において定義され且つ 確立されるようにRTIME/システムタイムアウトが発生した場合に装置によって設定
7	WERR	コンフィギュ レーション 書込エラー	(1=イベント;0=イベント無し)ホストによってアクセス不可能なレジスタ位置への 書込をホストが検出した場合に装置によって設定。このビットはロックレジスタの カーフィールドを介してホストアクセスがアンロックされる場合に設定されることはない。
30	SPURINT	偶発的CPC インタラプト	(1=イベント;0=イベント無し)無効な内部CPCインタラプトの受信を表すCPC ファームウェアによって設定。これは装置ハードウェア故障であり発生してはならない。
22	PPLOST	プロバイダ プリミティブ喪失	(1=イベント;0=イベント無し)ホストによって未だ読取られていないものの上に新た なプロバイダプリミティブをファームウェアが上書きする場合に装置によって設定。この 条件はプリミティブコマンドレジスタ(PCR)のPPAビットのテストにより検知される。
14	HPLOST	ホスト プリミティブ喪失	(1=イベント;0=イベント無し)ファームウェアによって未だ読取られていないもの の上に新たなホストプリミティブをホストが上書きする場合に装置によって設定。この 条件はプリミティブコマンドレジスタ(PCR)のHPAビットのテストにより検知される。
6	SPARE		
29,21 13,5	SDRIFT	ポートn統計 ドリフト	(1=イベント;0=イベント無し)対応するポート内の条件が統計停電が喪失される場 合のある状態に到達した場合にCPCファームウェアによって設定。これはポートの受 信FIFO内に使用可能な空間が無いためフレームが喪失するような受信輻輳が発生 している場合にのみ発生する。
28,20 12,4	ECN	ポートn早期 輻輳通知	(1=イベント;0=イベント無し)対応するポートの受信器における輻輳のアドバンス トホスト通知に対しCPCファームウェアによって設定。RxFIFO内の使用可能な空間が 無いため受信したフレームを落とすことをユニットが強制される場合に輻輳が発生する
27,19,11,3	FAN	フレーム アドレス通知	(1=イベント;0=イベント無し)フレームバッファ内にアドレスフィールドが存在する ことをホストに通知するために装置によって設定
26,18 10,2	SHL	ポートn統計 半寿命	(1=イベント;0=イベント無し)対応するポートの統計の内の1つ以上が、最後のアップ デートにより値性を変化(0から1又は1から0)する統計の最大桁ビットとして定義さ れる。半全マークをパスした場合にCPCファームウェアにより設定。
25,17 9,1	TINT	ポートn送信 インタラプト	(1=イベント;0=イベント無し)1つ以上のフレームの送信が完了した場合に装置に より設定。送信に成功したフレームの場合、このインタラプトは該フレームがチップを クリアしたことを知らせる。
24,16 8,0	RINT	ポートn受信 インタラプト	(1=イベント;0=イベント無し)対応するポートからホストシステムへ受信フレームが 完全に転送された場合に装置により設定。これはフレームがシステムメモリへ転送さ れたことを意味する。不良フレーム(SMRレジスタ内のRBUFF=0)の受信がデイス キューブルされると、不良フレームの場合にはRINTは発生されない。

【図10】

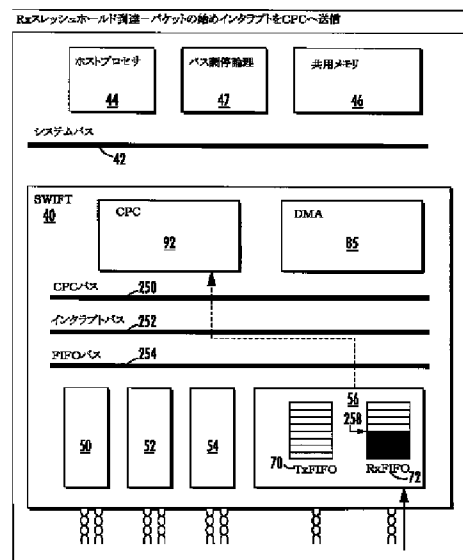
802.3データリンクレイヤヘッダ(18バイト)



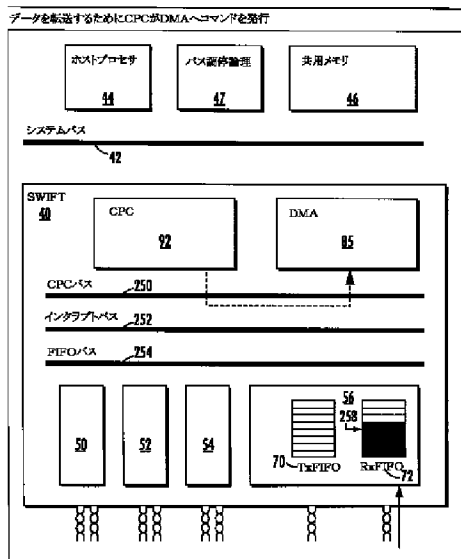
【図14】



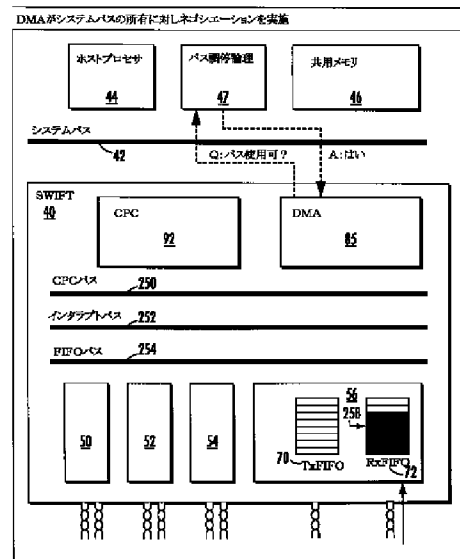
【図15】



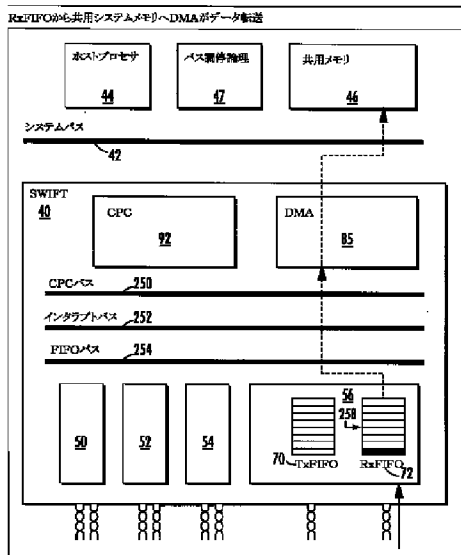
【図16】



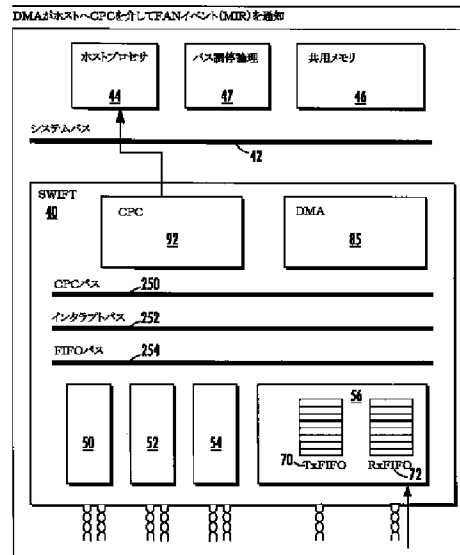
【図17】



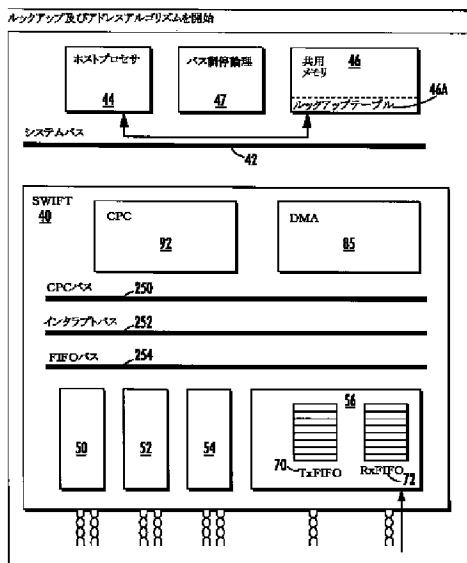
【図18】



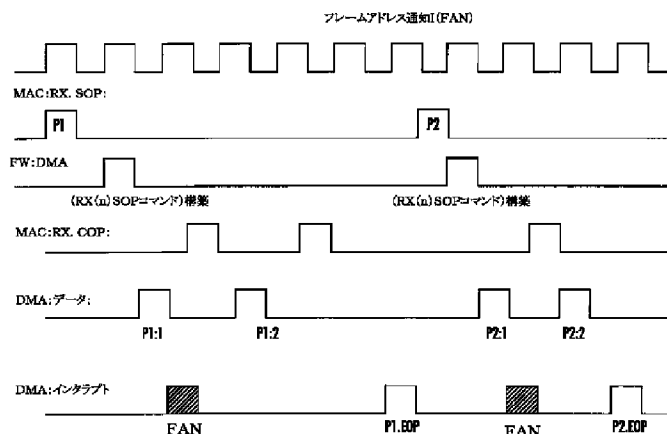
【図19】



【图 20】

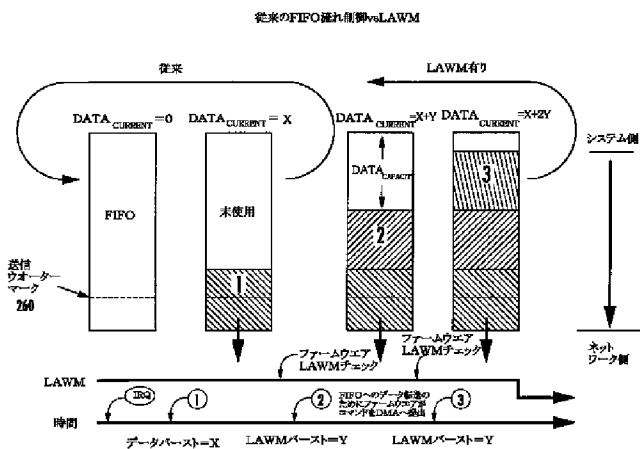


【图 2-1】

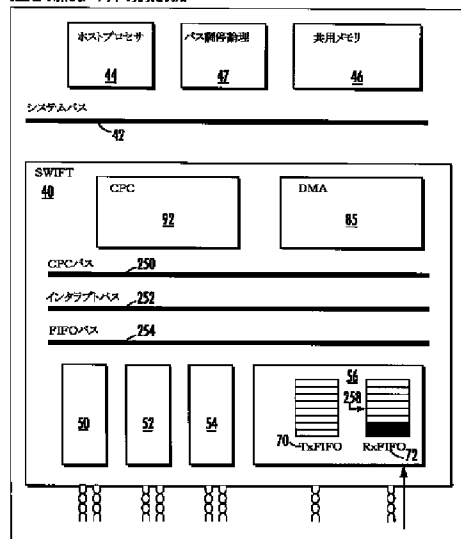


【图 29】

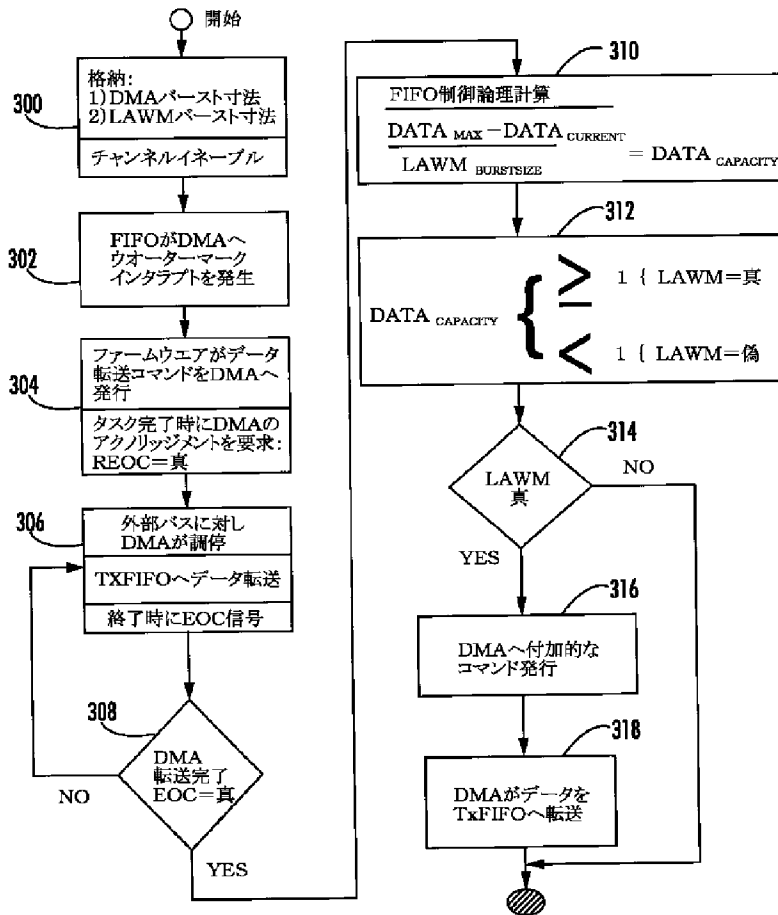
【图 2-2】



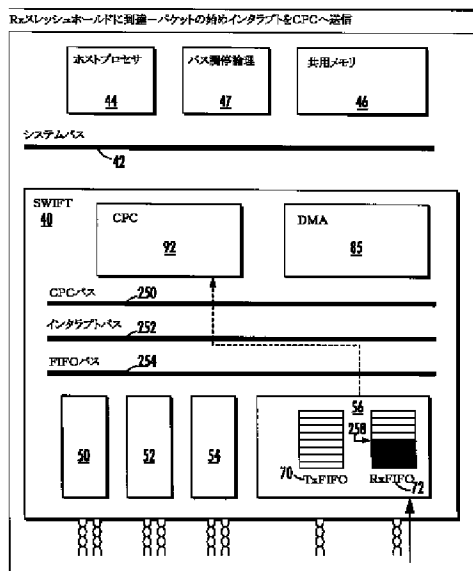
REFILFO新たなパッケージで充填を開始



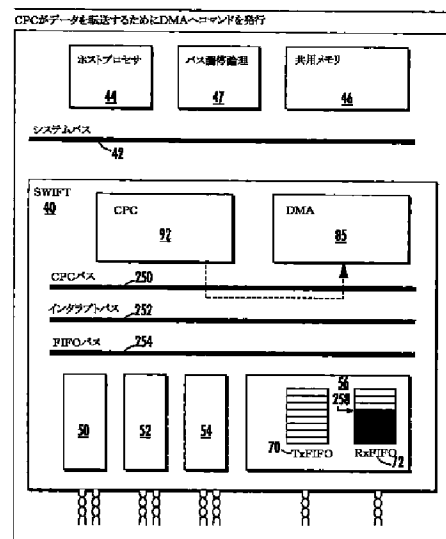
【図23】



【図30】

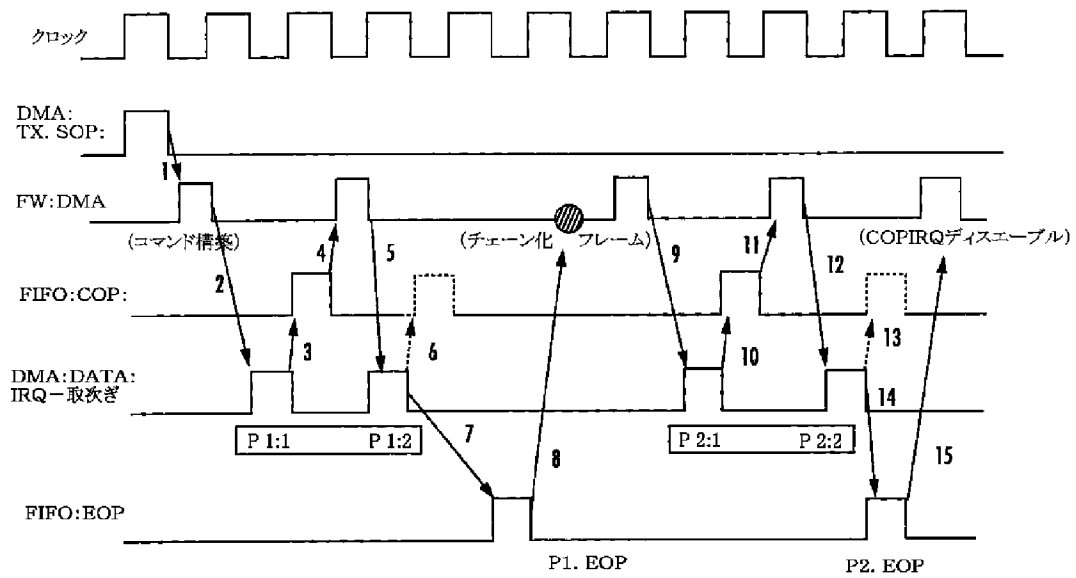


【図31】



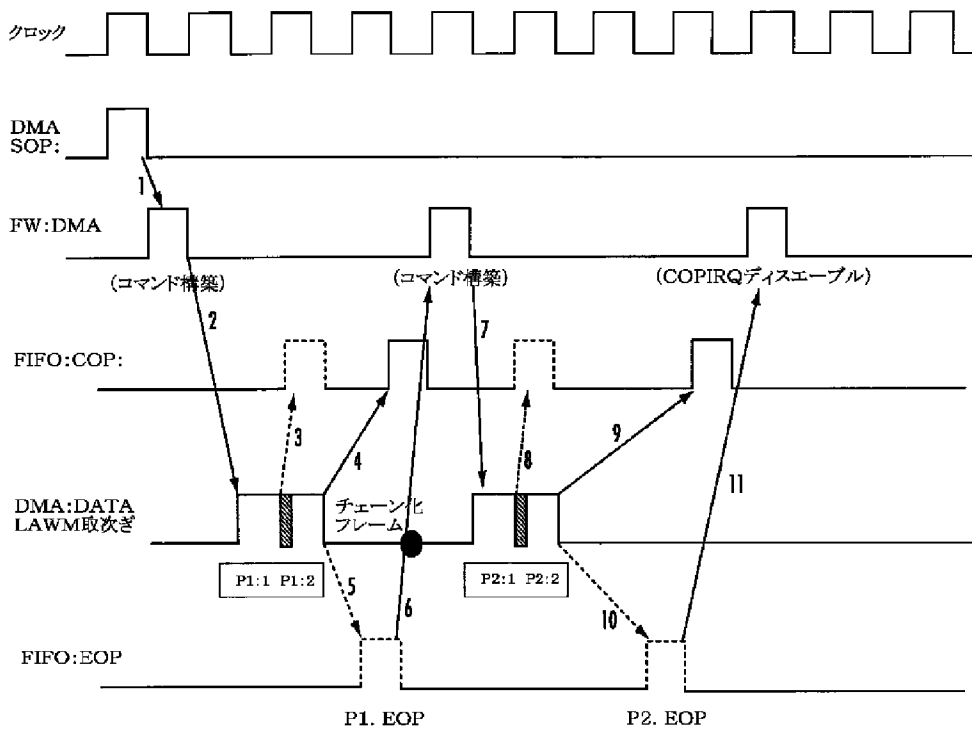
【図24A】

インタラプト取次ぎフレーム送信

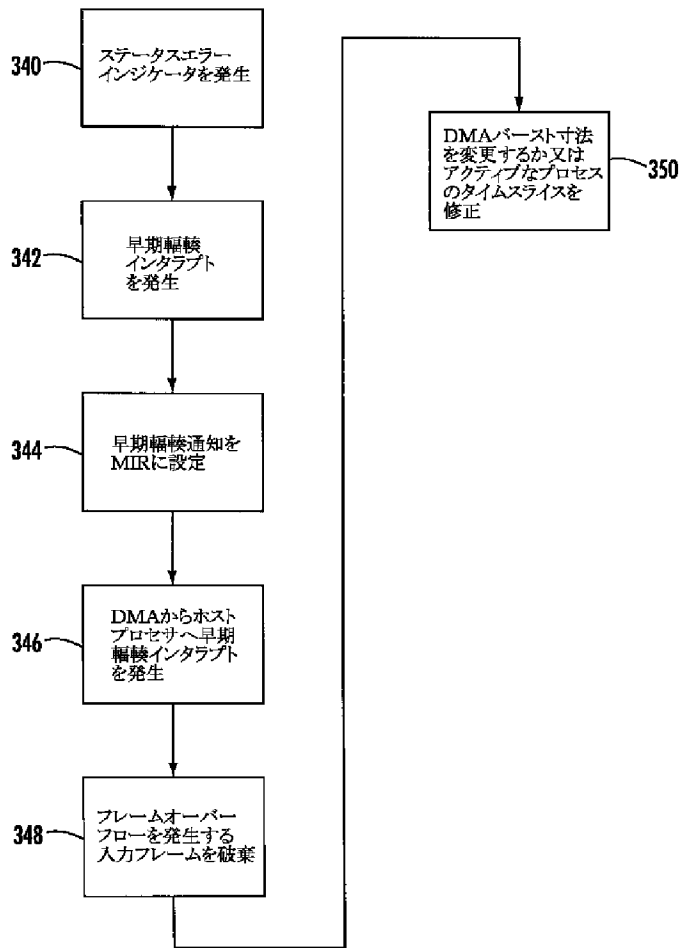


【図24B】

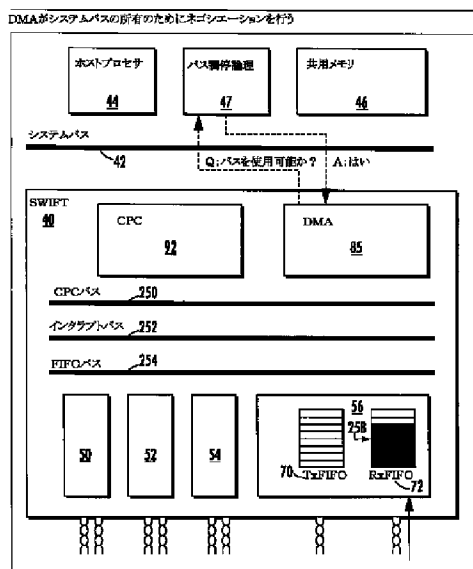
LAWM取次ぎフレーム送信



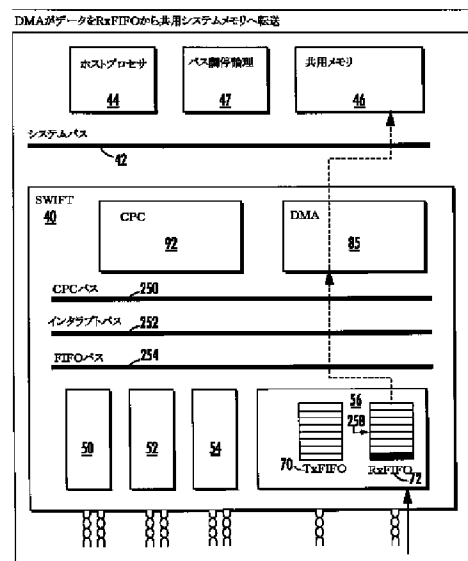
【図26】



【図32】

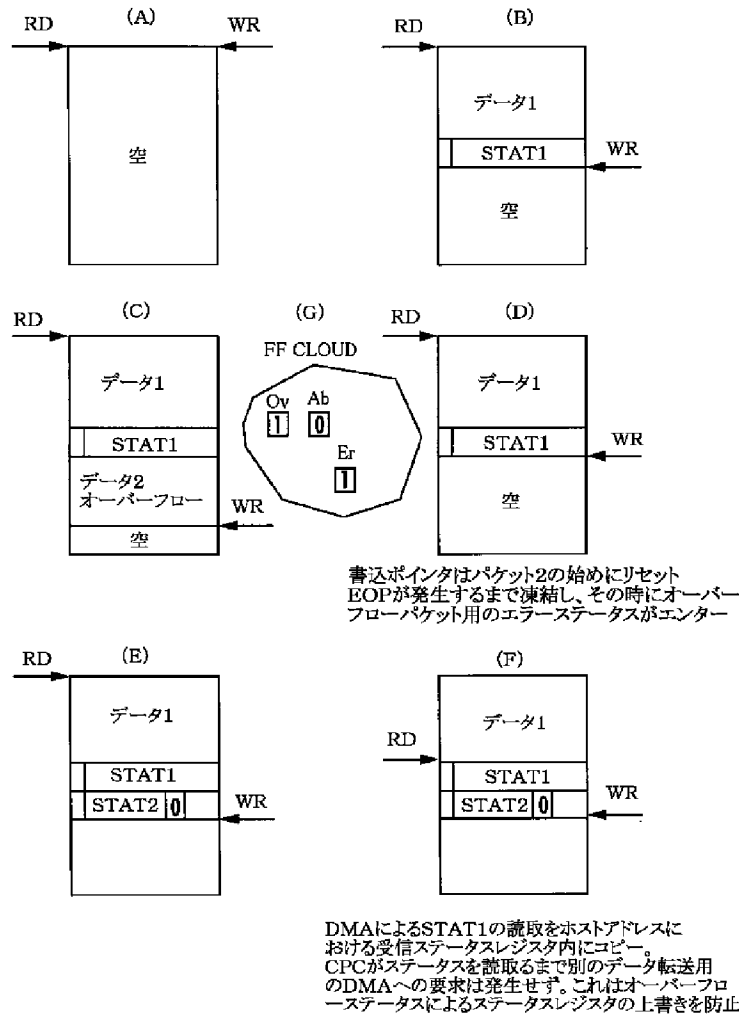


【図33】

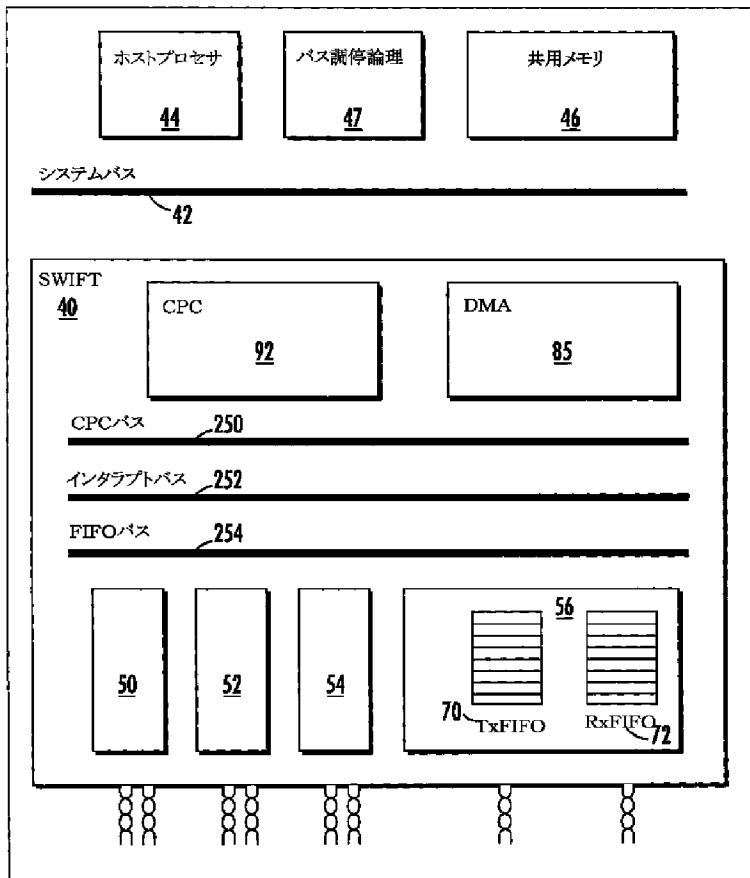


【図27】

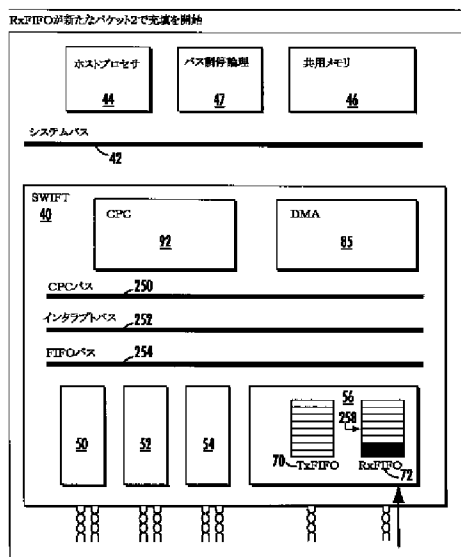
受信FIFOへの2番目のパケットでオーバーフローするFIFOの場合



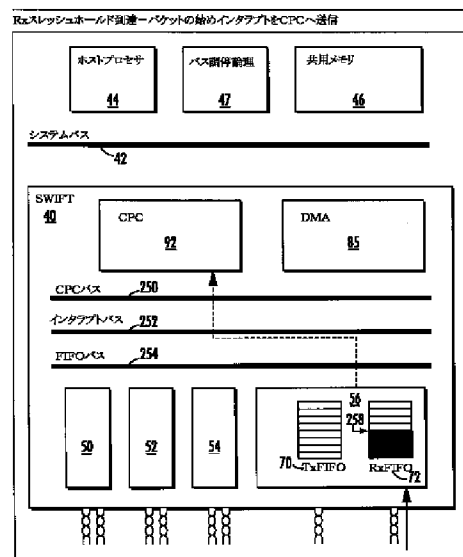
【図28】



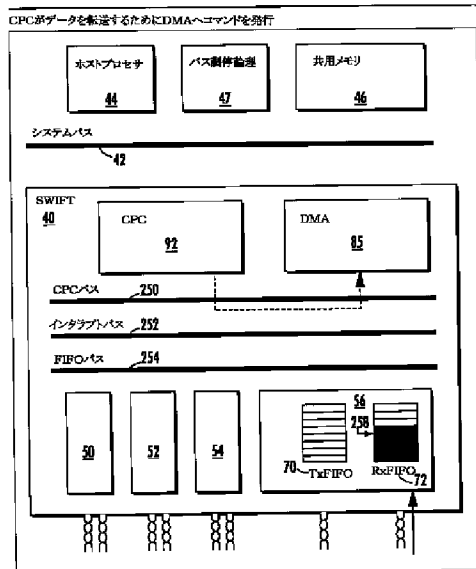
【図34】



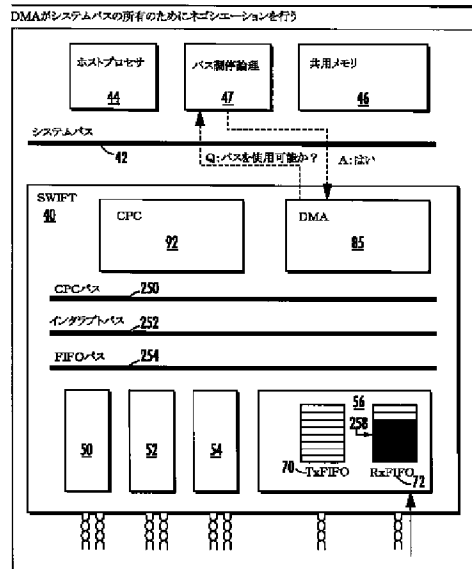
【図35】



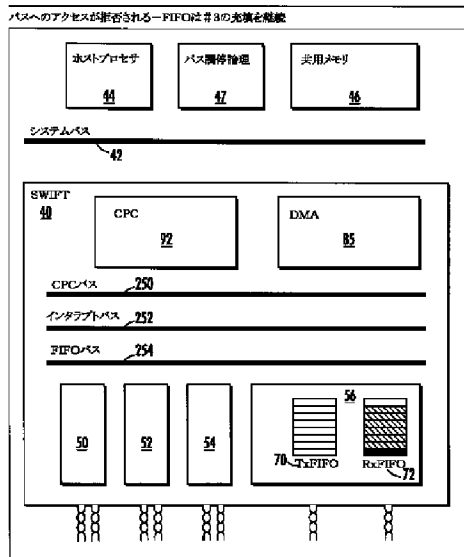
【図36】



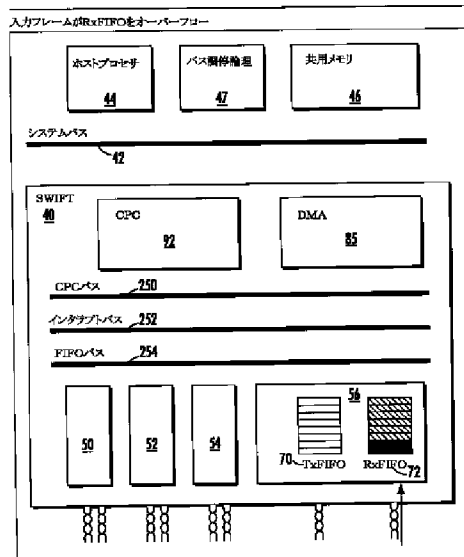
【図37】



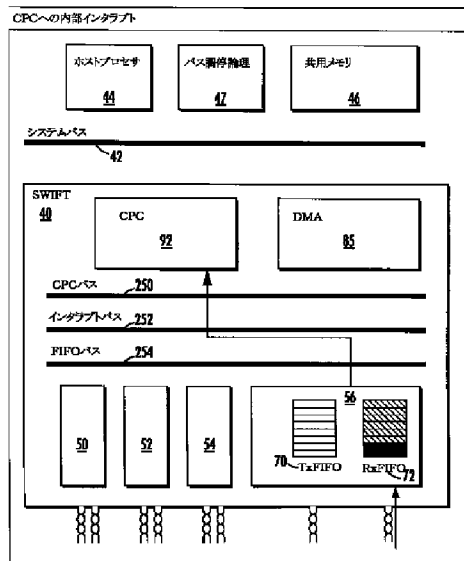
【図38】



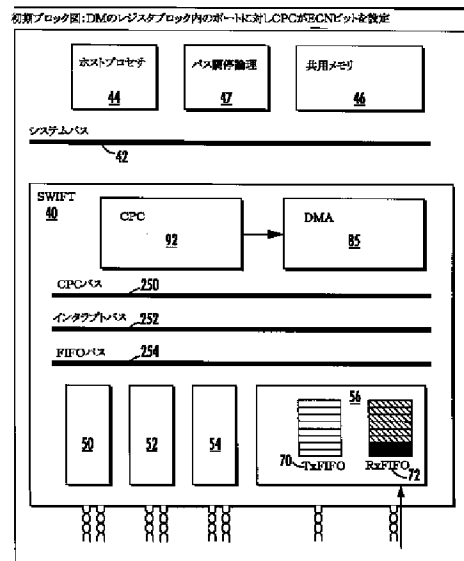
【図39】



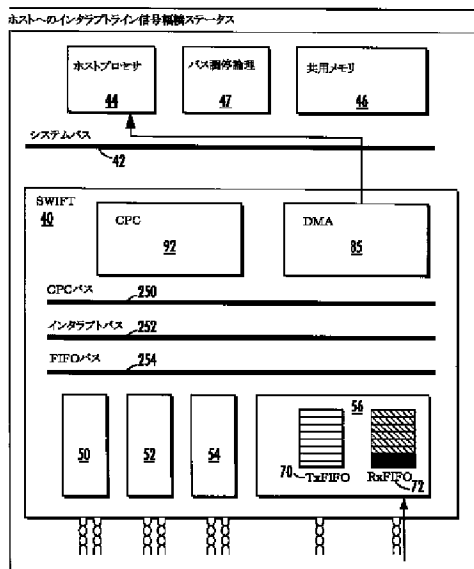
【図40】



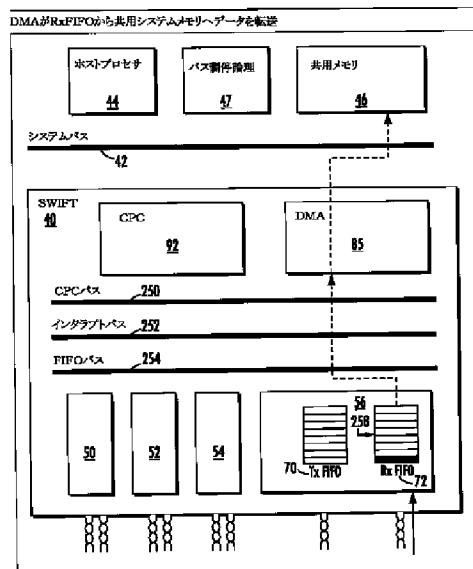
【図41】



【図42】



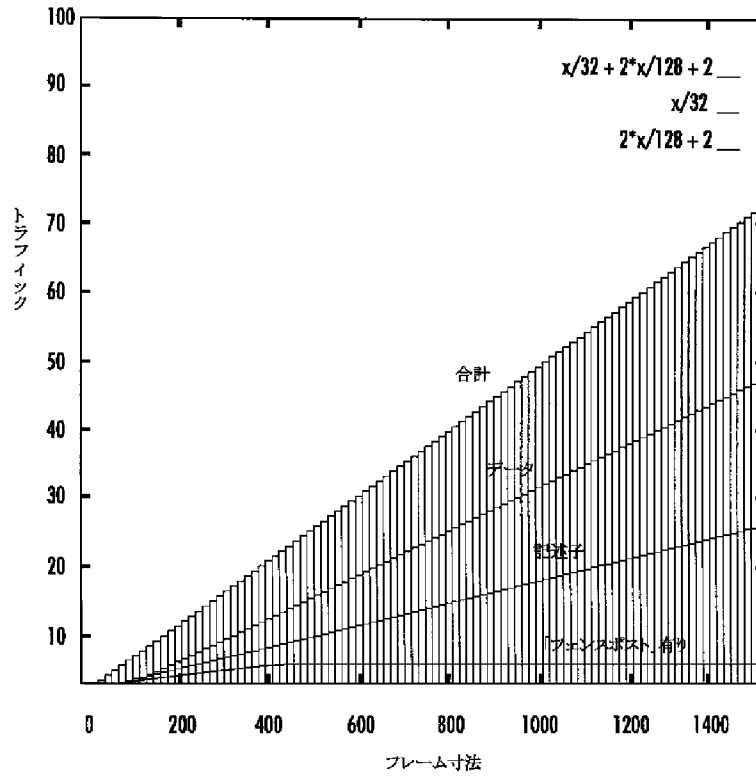
【図43】



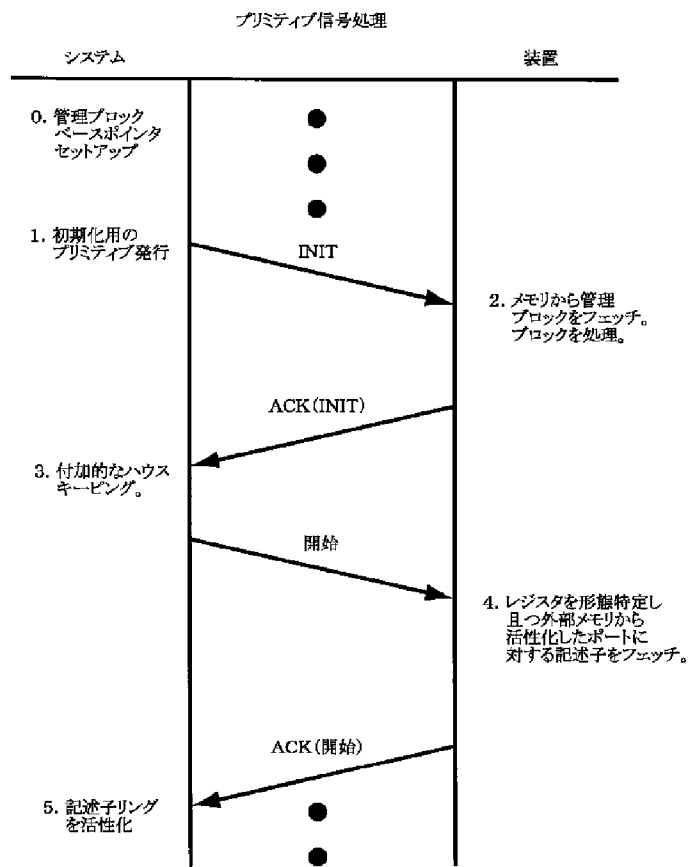
【図44】

ホストバスの推定トラフィック構成

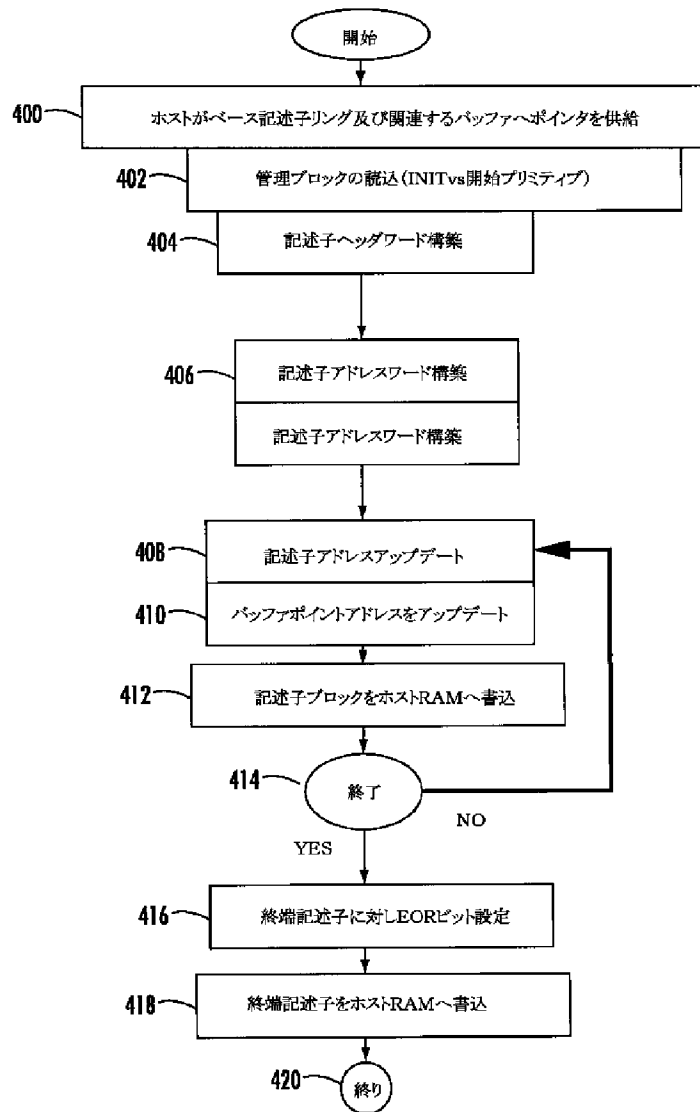
データ: $X/32$
 記述子: $2 * X/128 + 2$
 合計: $X/32 + 2 * X/128 + 2$



【図 4 5】



【図46】



【图 4-7】

メッセージ記述子0を受信

RMDG	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名称	OWN	EOR	ENP	BSIZE[12:2]												RERR	ROFLO	MSIZE[12:0]														

ビット#	フィールド	名称	記述
31	OWN	記述子 所有権	(1=装置;0=ホスト)受信メッセージ記述子及びその関連するデータバッファの所有を確立。所有ビットは装置とホストとの間ハビュクとして使用。所有が放棄されると、それと関連するバッファが変更されるべき場合に受信メッセージ記述子又は内容のどの部分も無い。
30	EOR	リング の終り	(1=リングの終り;0=リングの終りではない)記述子リング内の最後の受信メッセージ記述子を示す。この記述子を使用した後に装置をリングの一番上へリターンさせる。即ち、装置により使用される次の記述子がリング内の最初のエントリとなる。
29	ENP	フレーム の終り	(1=フレームの終り;0=チェン)関連する受信バッファが受信したフレームの終りを有することを表す。ゼロのENPは受信フレームが2個以上の隣接する記述子にスパンする場合のバッファの「チェン化」を意味する。
28:16	BSIZE	バッファ 寸法	(10ビット符合無し整数)関連する受信データバッファにおいて使用可能なバイト数を表す(最大8Kバイト)。BSIZEフィールドはビット12乃至2として定義されているのでバッファは4バイト(1ワード)毎にインクリメントすることに注意。BSIZEフィールドはこの定義を容易とさせるために上半分のワードに位置されている。ゼロのBSIZEは1ワードのバッファ寸法に対するデフォルトである。 バッファ内における実際の使用可能なバイト数はBSIZEフィールドとバッファの開始アドレス(RBADDR)によって決定される。受信データバッファは任意のバイトアドレスで開始することが許容されるが、常にワード整合した境界上で終了すべく装置により仮定される。即ち、全ての受信バッファの最後のアドレスは完全な4バイトワードである。
17	RERR	Rxエラー 要約	(1=エラー;0=正規)受信ステータスにおいて報告されるエラーステータスビットの論理OR要約がバッファ内のフレームの終りに続く最初の完全なワード内に装置によって書込まれる。RERR要約:CNTOVL, FCS, RABRT, ROFLO, 受信フレームに関連するエラーに対する単一ビットテストを可能とする。
16	ROFLO	RxFIFO オーバー フローエラー	(1=エラー;0=正規)受信FIFO内の使用可能な空間が不十分であるために落とされたパケットを表す。オーバーフローが発生すると、HDLCユニットは統計目的のために入力パケットのモニタを継続し、且つ全パケット(又は少なくとも未だFIFOから読取っていない部分)を落とす。その結果のステータスワードはパケットの終りタグと共にFIFO内に書込まれる。オーバーフローはFIFOの不適切なサービス(競取)により発生される。このビットがセットされると、MSIZEはバッファ内の実際のデータの量を表さない場合がある。
15:0	MSIZE	メッセージ 寸法	(15ビット符合無し整数)関連するバッファ内の受信フレームの一部又は全部によって占有されるオクテットの数を表す。MSIZEはバッファ内のフレームの終りに続く最初の完全なワード内に装置によって書込まれた受信ステータスワードの4個のオクテットを包含するものではない。 MSIZEフィールドは、ホストが記述子の所有を装置へ与える場合には全てゼロであると期待される。装置はこれをチェックしようとしないので、与えられたゼロでない全ての値は誤ったMSIZEをリターンすることになる。

【図48】

受信メッセージ記述子1

RMD1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名称	RBADR[31:0]																															

ビット#	フィールド	名称	記述															
31:0	RBADR	受信バッファ 開始アドレス	(32ビット符号無し整数)関連する受信データバッファの最初のアドレス位置へのポインタとして作用。受信データバッファは入力フレームを格納するために装置によって使用される。1個を超えることのないフレームが与えられたバッファ内に格納される。その寸法がバッファ寸法を超える場合には、単一のフレームが複数のバッファをスパンする場合がある。															
			RBADRは、受信バッファがワード整合境界上で開始することを要求されないことを意味する32ビットデータワードシステム内のバイトアドレスである。装置によって課される規則は、受信バッファは任意のバイト整合で開始することが可能であるが、常にワード整合境界上で終了することである。以下の表は2個の最小桁RBADRビットによって表されるバイト整合の概略である。															
			<table><tr><th>RBADR[1:0]</th><th>有効バイト</th><th>整合</th></tr><tr><td>00</td><td>4</td><td>整合(完全なワード)</td></tr><tr><td>01</td><td>3</td><td>不整合</td></tr><tr><td>10</td><td>2</td><td>不整合</td></tr><tr><td>11</td><td>1</td><td>不整合</td></tr></table>	RBADR[1:0]	有効バイト	整合	00	4	整合(完全なワード)	01	3	不整合	10	2	不整合	11	1	不整合
			RBADR[1:0]	有効バイト	整合													
			00	4	整合(完全なワード)													
01	3	不整合																
10	2	不整合																
11	1	不整合																

【図50】

送信メッセージ記述子1

TMD1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名称	TBADR[31:0]																															

ビット#	フィールド	名称	記述		
31:0	TBADR	送信バッファ 開始アドレス	(32ビット符合無し整数) 関連する送信データバッファの最初のアドレス位置へのポインタとして機能する。送信データバッファは出力フレームに対する発信元として装置により使用される。与えられたバッファ内には1つを超えるフレームを格納すべきではない。その寸法がバッファ寸法を超える場合には単一のフレームは複数のバッファをスパンすることが可能である。		
			TBADRは送信バッファがワード整合境界上で開始することが必要とされるものではないことを意味する32ビットデータワードシステム内のバイトアドレスである。以下の表は2個の最小桁TBADRビットによって表されるバイト整合の概略である。		
			TBADR[1:0]	有効バイト	整合
			00	4	整合(完全なワード)
			01	3	不整合
10	2	不整合			
11	1	不整合			

【図49】

送信メッセージ記述子0

TMD0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名称	OWN	FOR	ENP	NOCRC	TOFLO	予約済み											UFLO	MSIZE[12:0]														

ビット#	フィールド	名称	記述
31	OWN	記述子 所有権	(1=装置;0=ホスト)ホストによりセットされ、装置によりクリアされる。送信メッセージ記述子及びそれと関連するデータバッファの所有を確立する。所有ビットが装置とホストとの間のハンドシェイクとして使用される。所有が放棄された場合には、送信メッセージ記述子又は関連するバッファの内容のどの部分も変更されるべきではない。
30	FOR	リング の終り	(1=リングの終り;0=リングの終りではない)記述子がリング内の最後のエントリとしてマーク付けすべくホストにより形態特定される。記述子リング内の最後の送信メッセージ記述子を示す。この記述子を使用した後に装置をリングの一番上へリターンさせる。即ち、装置により使用される次の記述子はリング内の最初のエントリである。
29	ENP	フレーム の終り	(1=フレームの終り;0=フレームの終りではない)関連する送信データバッファが送信フレームの終りを包含していることを示すべくホストによりセットされる。ゼロのENPは送信されるべきフレームが2個以上の隣接する記述子をスパンする場合のバッファの「チェーン化」を意味する。
28A	NOCRC	CRC 添付無し	(1=非添付;0=添付)フレーム毎にTxCRC発生を制御することをホストにより形態特定される。フレームチェックシーケンス(CRC)がユニットによって自動的に発生され且つ添付されることを防止する。NOCRCはフレームの終り(ENP)ビットがセットされた場合に装置によって使用されるに過ぎない。
27	TOFLO	TxFIFO オーバーフロー エラー	(1=エラー;0=正規)FIFO Txがオーバーフローにある場合に、HDLCによってセットされる。多分リオーターマーク<パース寸法であるため。これは、FIFO Tx内の使用可能な空間を超えて書き込み使用したことを意味する。この条件から出る唯一の方法はTx FLUSH又はリセットをセットすることである。
26:17	RESERVED	—	ゼロでなければならない。
16	UFLO	TxFIFO アンダーフロー エラー	(1=エラー;0=正規)フレームの終りに遭遇する前に送信期間中に送信FIFOが空にされる場合に装置によってセットされる。アンダーフローがFIFOの不適切なサービス(書込)により発生される。
15:0	MSIZE	メッセージ 寸法	(13ビット符合無し整数)関連する送信データバッファ内に含まれる送信フレームのオクテットの数を表すべくホストによって設定される。